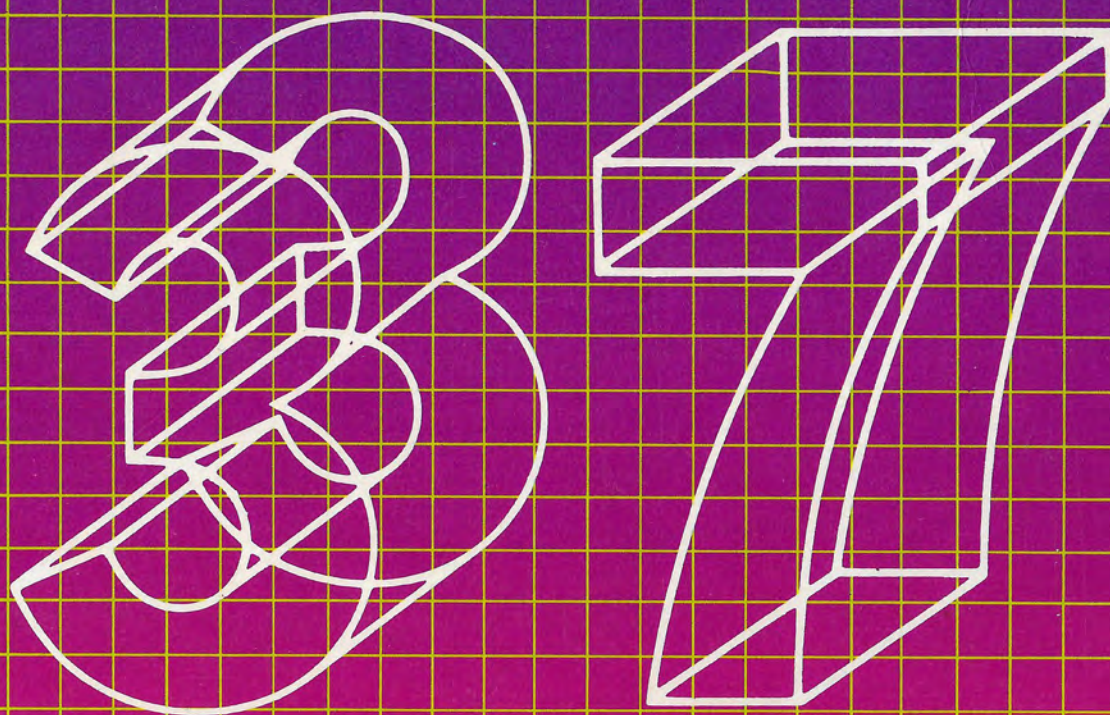


BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



DISFRUTA TU ORDENADOR

APRENDE Y HAZ COSAS CON EL

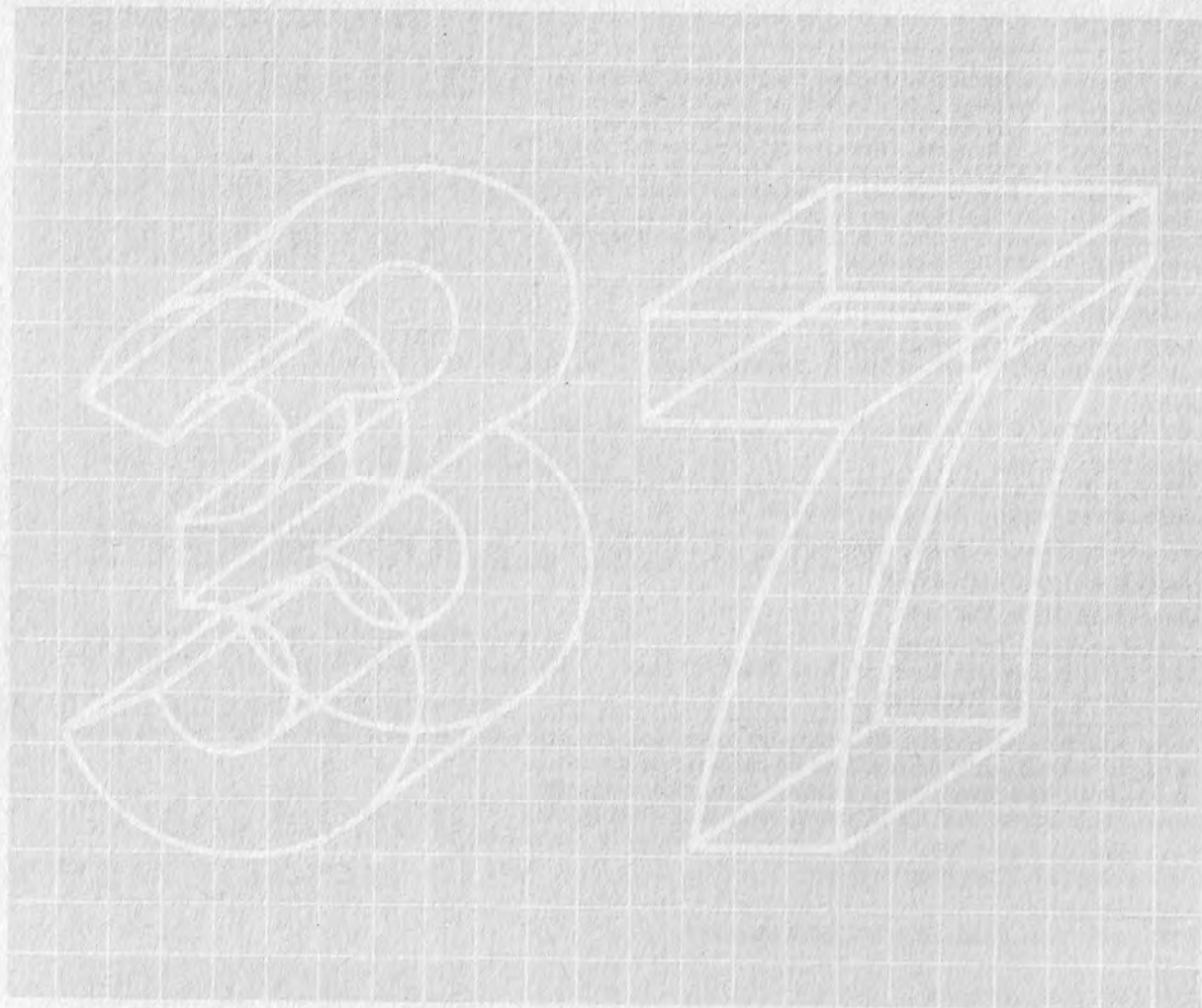
EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:

RICARDO ESPAÑOL CRESPO.

Gerente:

ANTONIO G. CUERPO.

Directora de producción:

MARIA LUISA SUAREZ PEREZ.

Director de la colección:

JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:

BRAVO-LOFISH.

Maquetación:

D. S. M.

Dibujos:

JOSE OCHOA

Tomo 37. «Experiencias y prácticas en logo», Carlos Albert, Técnico de Informática; Adoración Llena Jubelo, Técnica de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMÁTICA APLICADA: Francisco Blanco, Diplomado en Telecomunicación; Alejandro Marcos, Licenciado en Química, «El Taller de Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:

Pedro Teixeira, 8-2.º, Teléf. 455 09 99. 28020 Madrid.

Publicidad:

Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:

COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.

Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:

CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.

Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-177-0

ISBN de la obra: 84-7688-047-2

Fotocomposición:

ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:

GRAFICAS EMA. (Madrid)

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986

Printed in Spain - Impreso en España.

Suscripciones y números atrasados:

Ediciones Siglo Cultural, S. A.

Pedro Teixeira, 8-2.º. Teléf. 259 73 31. 28020 Madrid.

Diciembre, 1987

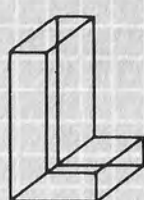
P.V.P. Canarias: 355.—Ptas.

INDICE

■ EXPERIENCIA Y PRACTICAS EN LOGO	5
■ MANEJO DE SPRITES Y ELEMENTOS GRAFICOS	15
■ TRUCOS Y RUTINAS BASICAS	23
■ EL TALLER DEL HARDWARE	33
■ APRENDER CON EL ORDENADOR	41
■ PEQUEÑA HISTORIA DE LA INFORMATICA	49
■ TEMAS MONOGRAFICOS DE VANGUARDIA	53
■ TERMINOLOGIA	57

EXPERIENCIA Y PRACTICAS EN LOGO

Otras primitivas



AS primitivas que explicamos a continuación sólo las encontramos en algunas versiones de Logo para los PC-COMPATIBLES.

— Pantalla

Devuelve un número que representa el número

de pantalla utilizada.

Si sólo hay una pantalla, devuelve un 1.

Si hay dos pantallas, una color y otra monocroma, devuelve un 2.

NOTA: Si utilizamos más de una pantalla, no podemos utilizar el comando PM.

? ES PANTALLA

1

Utilizando una sola pantalla.

? ES PANTALLA

2

Utilizando dos pantallas.

PONPANTALLA n

Fija el texto y el gráfico sobre una o dos pantallas, dependiendo del valor de n

Si n es 1, fija el texto y el gráfico sobre una sola pantalla.

Si n es 2, fija el texto sobre una pantalla y el gráfico sobre la otra.

Si disponemos solamente de una pantalla y tecleamos:

?PONPANTALLA 2

El LOGO mostrará el mensaje de error: PERIFERICO NO ACCESIBLE

NOTA: Si estamos utilizando dos pantallas, la primitiva PM no tendrán ningún efecto.

— PONCOM n velocidad paridad bits

stop

Esta primitiva establece la transmisión del puerto serie.

Para utilizar esta primitiva debes disponer de un adaptador de comunicaciones asíncronas, si no, el Logo mostrará el mensaje de error:

PERIFERICO INACCESIBLE.

Los datos que indicamos con esta primitiva pueden ser:

n 1 ó 2

Indica el número del adaptador de comunicaciones asíncronas.

velocidad 110, 150, 300, 600, 1200, 2400, 4800 ó 9600 baudios.

Indica la velocidad de transmisión.

paridad 0, 1 ó 2

0 no hay paridad

1 impar

2 par

bits 7 u 8

stop 1 ó 2

Indica el número de "stop bits"

Esta primitiva la utilizamos para enviar información, a través de la salida serie, al dispositivo conectado a ella, por ejemplo, a una impresora.

Gestión de la salida serie en el Spectrum

— ENTRADASERIE

Entrada serie (interface RS232).

Lee todo lo que llega a la entrada serie, define la tasa de baudios y devuelve un byte entre 0 y 255.

La unidad central puede soportar varios monitores.

EXPERIENCIA Y PRACTICAS EN LOGO

- .SALIDASERIE n
Salida serie (interface RS232).
Envía un byte a la salida serie.

Cuadro resumen

- PANTALLA
Devuelve un número que representa el número de pantalla utilizada.
Si hay una pantalla, devuelve un 1; si hay dos pantallas, devuelve un 2.
- PONPANTALLA n
Fija el texto y el gráfico sobre una o dos pantallas, dependiendo del valor de n
Si n es 1, fija el texto y el gráfico sobre pantalla.
Si n es 2, fija el texto sobre una pantalla y el gráfico sobre la otra.
- PONCOM n velocidad paridad bits stop.
Establece la transmisión del puerto serie.
- .ENTRADASERIE
Lee todo lo que llega a la entrada serie, define la tasa de baudios y devuelve un byte entre 0 y 255.
- .SALIDASERIE n
Envía un byte a la salida serie.

MUSICA I

Antes de empezar

Antes de introducirnos en la creación de música con el Logo, es conveniente recordar y repasar unos cuantos conceptos básicos que nos faciliten su comprensión.

La música se origina como consecuencia de una serie de vibraciones. Estas vibraciones tienen diferentes frecuencias y duración y crean lo que llamamos nota musical. Cada una de estas notas produce un tono diferente.

Las notas musicales podemos agruparlas en diferentes escalas. La más frecuente es la escala cromática, que está compuesta de 12 notas y son:

DO DO#
RE RE#

MI
FA FA#
SOL SOL#
LA LA#
SI

Las notas que van acompañadas del signo # son las que se denominan sostenidas y corresponden a tonos intermedios (semitonos).

Estos semitonos se llaman sostenidos si toman el nombre de la nota superior a ella y bemoles si toman el nombre de la nota siguiente a ella.

Un bemol se representa con el signo:

La escala que hemos visto anteriormente equivale a esta otra:

DO
RE $\flat$ RE
MI $\flat$ MI
FA
SOL $\flat$ SOL
LA $\flat$ LA
SI $\flat$ SI

Al conjunto de las siete notas principales, DO, RE, MI, FA, SOL, LA y SI se le llama octava, pudiéndose tocar estas notas en diferentes octavas.

En el teclado de un piano podemos observar con claridad lo que hemos explicado:



Fig. 1.

Las siete notas principales se pueden también expresar con la siguiente nomenclatura internacional:

Si utilizamos dos monitores podemos fijar uno para texto y otro para gráficos.

DO = C
 RE = D
 MI = E
 FA = F
 SOL = G
 LA = A
 SI = B

Las diferentes notas que se utilizan para componer una pieza musical se van representando sobre el pentagrama.

Un pentagrama es un conjunto de cinco líneas paralelas sobre las cuales o bien sobre los espacios que existen entre dos de ellas se dibuja la nota correspondiente.

Cada nota tiene un lugar fijo en el pentagrama:



Fig. 2.

Una pieza musical está compuesta por un conjunto de notas, pero cada una de estas notas puede tener una duración determinada y conseguir de esta forma lo que se llama ritmo.

La duración de una nota se representa en el pentagrama con diferentes signos. Estos signos son y se denominan:

○ = Redonda
 ♩ = Blanca
 ♪ = Negra
 ♫ = Corchea
 ♬ = Semicorchea
 ♭ = Fusa
 ♮ = Semifusa

Una duración determinada corresponde a la mitad de la anterior y al doble de la siguiente, es decir, una negra dura la mitad de una blanca y el doble de una corchea.

La duración más larga es la de la redonda y es la que se toma como referencia.

Si suponemos que la redonda tiene una duración de un segundo, podemos fácilmente calcular la duración de las demás:

○ = 1 segundo
 ♩ = 1/2 segundo
 ♪ = 1/4 segundo
 ♫ = 1/8 segundo
 ♬ = 1/16 segundo
 ♭ = 1/32 segundo
 ♮ = 1/64 segundo

De esta forma podemos obtener las siguientes equivalencias:

1 ○ = 2 ♩ = 4 ♪ = 8 ♫ = 16 ♬ = 32 ♭ = 64 ♮
 1 ♩ = 2 ♪ = 4 ♫ = 8 ♬ = 16 ♭ = 32 ♮
 1 ♪ = 2 ♫ = 4 ♬ = 8 ♭ = 16 ♮
 1 ♫ = 2 ♬ = 4 ♭ = 8 ♮
 1 ♬ = 2 ♭ = 4 ♮
 1 ♭ = 2 ♮

La duración más corta de una nota musical es la semifusa.

Otro elemento que forma parte de una composición musical es el silencio o la pausa. Estas pausas se pueden intercalar entre dos sonidos diferentes y haber más de una seguida.

Estas pausas o silencios tienen también diferentes duraciones.

La duración de cada nota tiene su correspondiente silencio

Su representación y denominación es la siguiente:

— = Silencio de redonda
 — = Silencio de blanca
 \ = Silencio de negra
 7 = Silencio de corchea
 7 = Silencio de semicorchea
 7 = Silencio de fusa
 7 = Silencio de semifusa

El silencio de mayor duración es el de la redonda y el de menos el de la semifusa.

La equivalencia de la duración de los silencios es igual a la de la duración de las notas:

La primitiva PM no tiene ningún efecto si utilizamos dos monitores.

EXPERIENCIA Y PRACTICAS EN LOGO

```

1 - 2 - 4 - 8 - 16 - 32 - 64
1 - 2 - 4 - 8 - 16 - 32
1 - 2 - 4 - 8 - 16
1 - 2 - 4 - 8
1 - 2 - 4
1 - 2

```

Una vez que hemos recordado algunos conceptos básicos y conocemos los elementos fundamentales que intervienen en la composición de una pieza musical (sonidos, duraciones y silencios), podemos estudiar la forma de interpretar todo esto con la ayuda del ordenador y del Logo.

El próximo tomo comenzaremos a verlo.

Ejercicios

1. Define los procedimientos necesarios para desarrollar el siguiente juego:

— En la pantalla aparecerán tres palabras elegidas al azar de una lista previamente definida.

— Tras unos segundos la pantalla se borrará y se permitirá al usuario introducir tres palabras.

— Si las palabras introducidas son iguales a las que aparecieron en un principio y mantienen el mismo orden, aparecerá en la fila correspondiente la palabra *CIERTO*; si no, aparecerá la palabra *FALSO*.

2. Intenta dibujar un boleto de la lotería primitiva

3. Define los procedimientos necesarios para dibujar sobre un pentagrama la nota que se introduzca desde el teclado con una duración de una negra.

4. Realiza el mismo proceso anterior, pero en esta ocasión con la posibilidad de elegir también la duración de cada nota.

5. ¿Son correctas las órdenes siguientes?

```

— ? PRIMITIVA? [AV 30]
— ? PARA A1
  > DEFINE "T1 [[] [AV 4 GD 90 RE 70]]
  > FIN
— ? ES DEFINIDO? "T1
— ? PARA A2
  > DEFINE "T2 PONULTIMO [AV 10

```

```
GD 90] TEXTOPROC "T1
```

```
> FIN
```

```
— ? PARA A3
```

```
> HAZ "B LISTA [SALON CON BAL-
CON] "ARMARIOS "Y "MOQUETA
```

```
> ES :B
```

```
> FIN
```

Solución de los ejercicios

1:

Para realizar este juego de memorización de palabras definimos los siguientes procedimientos:

LISTAS almacena en la variable A una lista (la que queramos) de palabras.

Nosotros hemos elegido 13 palabras.

```
? PARA LISTAS
```

```
> HAZ "A [CUADRO PUNTO QUESO
SUR AVE PINTOR
```

```
RAMA COCHE ACEITUNA RAYO
LEY SOLDADO ESTRELLA]
```

```
> FIN
```

El procedimiento ELIGE coge de la lista anterior tres palabras al azar.

```
? PARA ELIGE
```

```
> HAZ "N CUENTA :A
```

```
> HAZ "P1 (AZAR :N) + 1
```

```
> HAZ "P2 (AZAR :N) + 1
```

```
> HAZ "P3 (AZAR :N) + 1
```

```
> FIN
```

Palabras presenta en la pantalla las palabras seleccionadas en ELIGE, cada una en una fila distinta.

```
? PARA PALABRAS
```

```
> PONCT 3
```

```
> HAZ "K1 LISTA ELEMENTO :P1 :A
```

```
> HAZ "K2 LISTA ELEMENTO :P2 :A
```

```
> HAZ "K3 LISTA ELEMENTO :P3 :A
```

```
> PONCURSOR [1 10]
```

```
> ES :K1
```

```
> PONCURSOR [1 12]
```

```
> ES :K2
```

```
> PONCURSOR [1 14]
```

```
> ES :K3
```

```
> FIN
```

Algunas versiones de Logo disponen de primitivas que nos permiten fijar la transmisión del puerto serie.



Fig. 3.

El procedimiento PREGUN permite que se introduzcan desde el teclado y en tres filas distintas tres palabras.

- ? PARA PREGUN
- > PONCURSOR [15 10]
- > HAZ "L1 LL
- > PONCURSOR [5 12]
- > HAZ "L2 LL
- > PONCURSOR [15 14]
- > HAZ "L3 LL
- > FIN

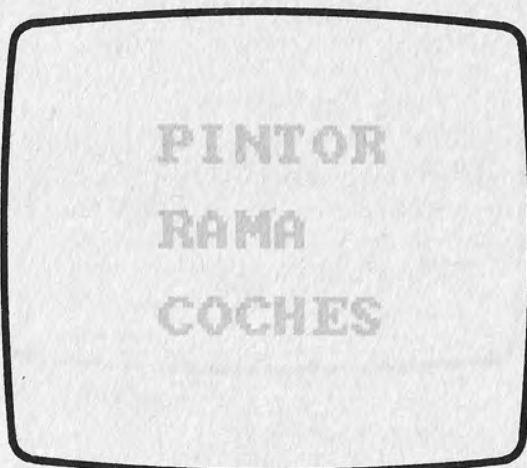


Fig. 4.

ACIERTOS analiza si la primera palabra introducida es igual a la primera palabra que se presentó en la pantalla; si es así, en la mis-

ma fila aparece la palabra CORRECTO, si no, aparece la palabra INCORRECTO, y así con las dos palabras restantes.

- ? PARA ACIERTOS
- > PONCT 2
- > PONCURSOR [27 10]
- > SI :K1 = :L1 [ES "CORRECTO] [ES "INCORRECTO]
- > PONCURSOR [27 12]
- > SI :K2 = :L2 [ES "CORRECTO] [ES "INCORRECTO]
- > PONCURSOR [27 14]
- > SI :K3 = :L3 [ES "CORRECTO] [ES "INCORRECTO]
- > FIN

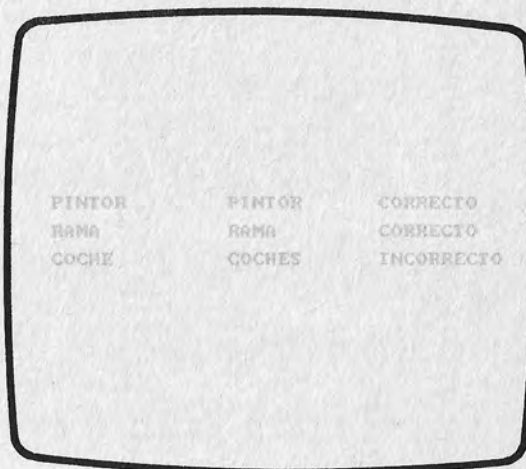


Fig. 5.

Finalmente el procedimiento A inicializa la pantalla y llama a los procedimientos anteriormente definidos.

- ? PARA A
- > PT
- > BT
- > LISTAS
- > ELIGE
- > PALABRAS
- > ESPERA 40
- > BT
- > PREGUN
- > PALABRAS
- > ACIERTOS
- > FIN

Una impresora puede ser el dispositivo que conectemos a la salida serie de nuestro ordenador.

EXPERIENCIA Y PRACTICAS EN LOGO

Empieza el juego ejecutando el procedimiento A.

? A

2:

Para realizar el dibujo del boleto definimos el siguiente procedimiento:

? PARA BOLETO

```
> OT
> BP
> BT
> SL
> PONPOS [-120 -64]
> BL
> PONCL 1
> PONRUMBO 0
> REPITE 2 [AV 150 GD 90 AV 240 GD 90]
> SL
> PONPOS [-50 -54]
> BL
> HAZ "R 3 * 48 - 6
> REPITE 8 [AV 105 GD 90 AV :R GD 90]
> PONCL 1
> REPITE 7 [GD 90 AV 6 GI 90 AV 105 RE 105]
> REPITE 2 [REPITE 6 [GD 90 AV 1 PONCL 2 GI 90 AV 105 RE 105] PONCL 1 REPITE 7 [GD 90 AV 6 GI 90 AV 105 RE 105]]
> GI 90
> PONCL 1
> REPITE 7 GD 90 AV 7 GI 90 AV :R RE :R]
> PONCL 2
> REPITE 7 [GD 90 AV 1 GI 90 AV :R RE :R]
> PONCL 1
> REPITE 7 [GD 90 AV 7 GI 90 AV :R RE :R]
> SL
> PONPOS [-105 -56]
> BL
> PONRUMBO 0
> REPITE 2 [AV 110 GD 90 AV 40 GD 90]
> AV 35
> GD 90
```

```
> AV 40
> HAZ "P -25
> REPITE 3 [SL PONPOS LISTA -102 :P BL AV 34 HAZ "P :P - 7]
> HAZ "P -97
> REPITE 4 [SL PONPOS LISTA :P -15 BL PONRUMBO O BL AV 65 HAZ "P :P + 8]
> SL
> PONPOS [-103 -54]
> BL
> PONRUMBO O
> REPITE 2 [AV 10 GD 90 AV 36 GD 90]
> PONCURSOR [6 5] ES [LOTERIA PRIMITIVA]
> SL
> PONPOS [35 60]
> BL
> REPITE 2 [AV 20 GD 90 AV 80 GD 90]
> PONCURSOR [25 5]
> ES "118540794
> FIN
```

Obtenemos el dibujo del boleto tecleando:

? BOLETO

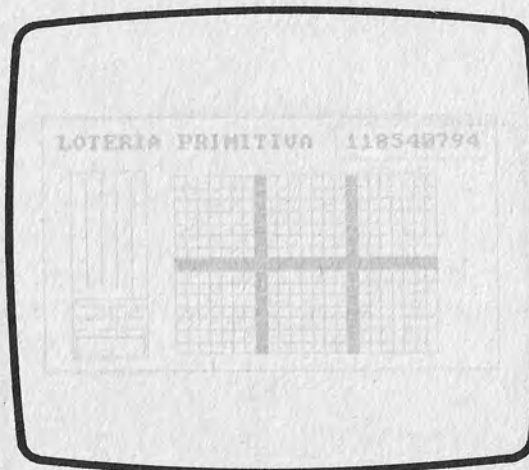


Fig. 6.

Si te resultan pesados los procedimientos tan largos, desglosa BOLETO en pequeños procedimientos y define un procedimiento principal que los vaya ejecutando.

El Logo dispone de primitivas para emitir sonidos. Casi todos los ordenadores llevan incorporado un altavoz.

3:

El primer procedimiento es el que dibuja el pentagrama:

? PARA PENTAGRAMA

- > SL
- > PONPOS [-100 10]
- > PONRUMBO 90
- > REPITE 5 [BL AV 200 GI 90
AV 1 GI 90 AV 200 GD 90 SL
AV 10 GD 90]
- > HAZ "INC 0
- > INTRO
- > FIN

INTRO es el que solicita la nota y controla de cuál se trata. Dependiendo de cuál sea se irá a dibujar una negra en el punto de coordenada Y que indica la variable I:

? PARA INTRO

- > PONCURSOR [2 20]
- > ES [INTRODUCE UNA NOTA:]
- > PONCURSOR [22 20]
- > PONCT FONDO
- > ES "..."
- > PONCT 7
- > PONCURSOR [22 20]
- > ES "..."
- > PONCURSOR [22 20]
- > HAZ "N LL
- > SI :N = [DO] [HAZ "I 0 NOTA]
- > SI :N = [RE] [HAZ "I 5 NOTA]
- > SI :N = [MI] [HAZ "I 10 NOTA]
- > SI :N = [FA] [HAZ "I 15 NOTA]
- > SI :N = [SOL] [HAZ "I 20 NOTA]
- > SI :N = [LA] [HAZ "I 25 NOTA]
- > SI :N = [SI] [HAZ "I 30 NOTA]
- > INTRO
- > FIN

Nota es la que dibuja la negra:

? PARA NOTA

- > SL
- > PONX -80 + :INC
- > PONY :I
- > BL
- > PONRUMBO 0
- > REPITE 10 [AV 2 GI 36]
- > SL
- > GI 90 AV 2
- > BL
- > RELLENA
- > RE 2 GD 90

> AV 22

> HAZ "INC :INC + 20

> FIN

Por último, englobamos en MUSICA todos los procedimientos:

? PARA MUSICA

- > BP
- > BT
- > OT
- > PENTAGRAMA
- > FIN

Para ejecutarlo introducimos:

? MUSICA

4:

En el procedimiento INTRO tenemos que realizar los siguientes cambios:

? PARA INTRO

- > PONCURSOR [2 17]
- > ES [INTRODUCE LA DURACION]
- > PONCT FONDO
- > PONCURSOR [25 17]
- > ES "..."
- > PONCURSOR [22 20]
- > ES "..."
- > PONCT 7
- > PONCURSOR [25 17]
- > ES "..."
- > PONCURSOR [25 17]
- > HAZ "D LL
- > PONCURSOR [2 20]
- > ES [INTRODUCE UNA NOTA:]
- > PONCURSOR [22 20]
- > ES "..."
- > PONCURSOR [22 20]
- > HAZ "N LL
- > PONCL 1
- > SI :D = [1] [REDONDA]
- > SI :D = [2] [BLANCA]
- > SI :D = [3] [NEGRA]
- > SI :D = [4] [CORCHEA]
- > SI :D = [5] [SEMICORCHEA]
- > SI :D = [6] [FUSA]
- > SI :D = [7] [SEMIFUSA]
- > INTRO
- > FIN

En INTRO se controla en este caso la duración, y dependiendo de cuál sea, se va a dibujar la nota correspondiente.

Un pentagrama puede ser traducido en una serie de números.

EXPERIENCIA Y PRACTICAS EN LOGO

INTRO1 controla la nota que se ha introducido:

```
? PARA INTRO1
> SI :N = [DO] [HAZ "I 0]
> SI :N = [RE] [HAZ "I 5]
> SI :N = [MI] [HAZ "I 10]
> SI :N = [FA] [HAZ "I 15]
> SI :N = [SOL] [HAZ "I 20]
> SI :N = [LA] [HAZ "I 25]
> SI :N = [SI] [HAZ "I 30]
> FIN
```

Para dibujar las notas con las diferentes duraciones definimos los siguientes procedimientos:

```
? PARA P1
> INTRO1
> SL
> PONX -80 + :INC
> PONY :I
> BL
> PONRUMBO 0
> REPITE 10 [AV 2 GI 36]
> FIN
```

```
? PARA REDONDA
> P1
> HAZ "INC :INC + 20
> FIN
```

```
? PARA BLANCA
> P1
> AV 22
> HAZ "INC :INC + 20
> FIN
```

```
? PARA NEGRA
> INTRO1
> SL
> PONX -80 + :INC
> PONY :I
> BL
> PONRUMBO 0
> REPITE 10 [AV 2 GI 36]
> SL
> GI 90 AV 2
> BL
> RELLENA
> RE 2 GD 90
```

```
> AV 22
> HAZ "INC :INC + 20
> FIN
```

```
? PARA CORCHEA
> P1
> SL
> GI 90 AV 2
> BL
> RELLENA
> RE 2 GD 90
> AV 22
> GD 90 AV 8
> HAZ "INC :INC + 20
> FIN
```

```
? PARA SEMICORCHEA
> P1
> SL
> GI 90 AV 2
> BL
> RELLENA
> RE 2 GD 90
> AV 22
> GD 90 AV 8
> RE 8 GD 90
> AV 3 GI 90
> AV 8
> HAZ "INC :INC + 20
> FIN
```

```
? PARA FUSA
> P1
> SL
> GI 90 AV 2
> BL
> RELLENA
> RE 2 GD 90
> AV 22
> GD 90 AV 8
> REPITE 2 [RE 8 GD 90 AV 3 GI 90 AV
8]
> HAZ "INC :INC + 20
> FIN
```

```
? PARA SEMIFUSA
> P1
> SL
```

*Con el Logo podemos componer piezas musicales.
Podemos transformar el teclado del ordenador
en un teclado musical.*

```

> GI 90 AV 2
> BL
> RELLENA
> RE 2 GD 90
> AV 22
> GD 90 AV 8
> REPITE 3 [RE 8 GD 90 AV 3 GI 90 AV
8]
> HAZ "INC :INC + 20
> FIN

```

Para ejecutar este proceso tendríamos que volver a introducir:

? MUSICA

5:

— ? PRIMITIVA? [AV 30]

INCORRECTO.

La primitiva PRIMITIVA? no acepta una lista como dato. El Logo devuelve el mensaje de error:

PRIMITIVA? NO ACEPTA [AV 30]
COMO DATO

— ? PARA A1

```

> DEFINE "T1 [[] [AV 4 GD 90 RE
70]]
> FIN

```

CORRECTO

Cuando ejecutemos A1 definiremos el procedimiento T1, que si lo llevamos al editor tendrá la forma:

? PARA T1

```

> AV 4 GD 90 RE 70

```

```

> FIN

```

— ? ES DEFINIDO? "T1

CORRECTO

El Logo devuelve CIERTO.

— ? PARA A2

```

> DEFINE "T2 PONULTIMO [AV 10
GD 90] TEXTOPROC "T1
> FIN

```

CORRECTO.

Cuando ejecutemos A2 se definirá el procedimiento T2, que será el procedimiento T1 más las órdenes [AV 10 GD 90]

? PARA T2

```

> AV 4 GD 90 RE 70
> AV 10 GD 90
> FIN

```

— ? PARA A3

```

> HAZ "B LISTA [SALON CON BAL-
CON] "ARMARIOS "Y "MOQUETA
> ES :B
> FIN

```

INCORRECTO.

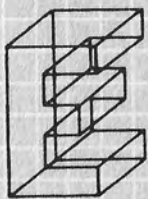
Si ejecutamos A3 en la pantalla aparecerá el mensaje:

NO SE QUE DEBO HACER CON Y

La primitiva LISTA si tiene más de dos datos debe ir junto con los datos entre paréntesis.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

Fantasmas en tu ordenador



L SPECTRUM es uno de los ordenadores menos agradados a la hora de realizar movimiento. Casi todos los demás ordenadores nos permiten usar Sprites o gráficos de distinto tamaño para relizar animación. En el

SPECTRUM, como hay que hacerlo desde BASIC, el resultado es muy lento y nada satisfactorio. Por ello, vamos a ver un programa en código máquina que nos permitirá tener hasta 36 Sprites a la vez en pantalla.

Lo primero que tenemos que hacer es introducir el programa l y ejecutarlo con RUN. Una vez hecho esto, el ordenador empezará a

pedirnos una serie de números. Estos números son el código máquina que nos permitirá mover Sprites por la pantalla. Lo único que hay que hacer es introducir, uno a uno, todos los números que aparecen en la figura 1.

1 COL = 1 FIL = 1	4 COL = 2 FIL = 1	7 COL = 3 FIL = 1
2 COL = 1 FIL = 2	5 COL = 2 FIL = 2	8 COL = 3 FIL = 2
3 COL = 1 FIL = 3	6 COL = 2 FIL = 3	9 COL = 3 FIL = 3

Fig. 1. Orden de definición.

```
1 DEF FN m(a,b,c,d)=USR 51312: DEF FN a(a,b)=USR 51131: DEF FN c(a)=USR 50000
DEF FN i( )=USR 51162: DEF FN b(a)=USR 51180: DEF FN d(a,b,c,d,e,f,g)=USR 51405
10 CLEAR 50999
11 FOR i=51000 TO 51740
12   INPUT "Numero: ";a
13   POKE i,a
14   LET chek=chek+a
15 NEXT i
16 IF chek<>80920 THEN GO TO 9000
20 CLS : PRINT "ROUTINA LISTA PARA SER USADA"
30 PRINT : PRINT : PRINT
40 PRINT "quieres grabarla? (sVn)"
50 LET a$=INKEY$
60 IF a$="" THEN GO TO 50
70 IF a$="s" OR a$="S" THEN SAVE "SPRUTES"CODE 51000,731
80 STOP
9000 CLS
9010 PRINT "HAS INTRODUCIDO MAL LOS NUMEROS"
9020 PRINT
9030 PRINT "INTENTALO OTRA VEZ"
9040 STOP
```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

221 42 11 92 221 126 4 79 205 1
202 205 14 200 229 221 225 221 126 2
7 7 7 87 221 134 0 103 221 126
3 7 7 7 95 221 134 1 111 58
111 200 167 202 38 201 221 33 18 202
71 121 8 14 1 221 126 2 167 32
17 221 35 221 35 221 35 221 35 221
35 221 35 221 35 12 24 233 7 7
7 221 134 0 148 186 56 7 40 5
237 68 186 48 20 221 126 3 7 7
7 221 134 1 149 189 56 13 40 11
237 68 187 56 6 16 200 1 0 0
201 8 185 32 3 8 24 243 6 0
201 221 42 11 92 221 126 4 205 1
202 205 14 200 221 70 12 229 221 225
197 205 40 201 193 221 112 4 205 40
201 201 175 50 111 200 33 18 202 17
19 202 1 245 0 54 0 237 176 201
221 42 11 92 221 126 4 205 1 202
205 14 200 229 221 225 58 141 92 221
119 4 205 40 201 221 54 2 0 33
111 200 53 201 167 202 38 201 254 37
210 38 201 84 93 19 19 26 167 192
33 73 200 229 62 1 205 1 22 6
19 225 126 215 35 16 251 237 75 69
92 205 27 26 62 58 215 253 78 13
6 0 205 27 26 253 203 1 110 40
250 207 255 83 112 114 105 116 101 32
110 111 32 100 101 102 105 110 105 100
111 32 83 112 114 105 116 101 32 121
97 32 100 101 102 105 110 105 100 111
32 0 221 42 11 92 221 126 4 205
1 202 205 14 200 229 221 126 12 254
192 210 38 201 71 35 35 126 7 7
7 128 254 192 210 38 201 221 126 20
79 35 126 7 7 7 129 218 38 201
221 86 29 221 94 28 221 225 221 126

4 245 213 197 58 141 92 221 119 4
205 40 201 193 209 241 221 119 4 221
112 0 221 113 1 221 115 5 221 114
6 205 40 201 201 221 42 11 92 221
126 4 205 1 202 84 93 19 19 26
167 40 6 33 92 200 195 33 200 229
221 126 12 119 71 221 126 20 35 119
79 254 192 48 49 35 221 126 28 119
7 7 7 128 254 192 48 36 35 221
126 36 119 7 7 7 129 56 25 35
221 126 44 119 221 126 52 35 119 221
126 53 35 119 221 225 205 40 201 33
111 200 52 201 207 10 221 126 4 8
221 70 2 221 78 3 221 102 0 221
110 1 81 92 213 125 230 7 245 214
7 237 68 87 135 130 50 189 201 205
212 201 221 86 6 221 94 5 121 72
245 229 197 197 229 205 244 201 8 119
8 225 6 8 26 205 185 201 174 119
121 44 174 119 45 205 170 201 19 16
239 193 16 225 193 225 44 65 241 61
194 84 201 65 17 32 0 205 244 201
241 40 11 8 119 25 18 252 8 167
193 4 24 12 235 38 0 104 41 41
41 41 41 25 45 193 121 230 7 200
8 119 45 18 252 201 36 124 230 7
192 62 32 133 111 216 124 214 8 103
201 14 0 167 24 251 31 203 25 31
203 25 31 203 25 31 203 25 31 203
25 31 203 25 31 203 25 201 84 93
122 230 192 31 31 31 246 64 103 122
230 56 23 23 111 122 230 7 180 103
123 203 63 203 63 203 63 133 111 201
124 230 24 203 47 203 47 203 47 198
88 103 201 33 18 202 22 0 95 8
29 123 7 7 7 147 95 25 8 201
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

Hay que tener mucho cuidado para no equivocarnos al introducir los números. De todas formas, en caso de que lo hagamos, el ordenador nos avisará.

Una vez que hemos metido todos los números sin problemas, el ordenador nos pregunta si queremos grabar el programa en có-

digo máquina. Lo normal es contestar que sí y almacenarlo en una cinta de cassette que esté limpia. Se recomienda dejar que pase la cinta durante dos o tres minutos sin grabar nada y, luego grabar la rutina. Esto es porque antes de la rutina tendremos que grabar el programa 2.

```

1 DEF FN m(a,b,c,d)=USR 51312: DEF FN a(a,b)=USR 51131: DEF FN c(a)=USR 50000
: DEF FN i( )=USR 51162: DEF FN b(a)=USR 51180: DEF FN d(a,b,c,d,e,f,g)=USR 51405
10 CLEAR 50999: LOAD ""CODE 51000,731
20 CLS : PRINT "ROUTINA LISTA PARA SER USADA"

```

Este segundo programa tiene en la línea número 1 todas las definiciones necesarias para poder utilizar las rutinas de Sprite. La línea 10 se encarga de hacer el CLEAR necesario para alojar a la rutina y de leerla y al-

macenarla en memoria. La línea 20 nos dice que la carga se ha efectuado sin problemas, por lo que no sirve para mucho y se puede borrar.

La importancia de este programa resi-

de en la línea 1. Esta línea tendrá que estar incluida en todos los programas en los que utilizemos Sprites.

Para todos aquellos a los que les interese el listado fuente de la rutina, aparece a continuación:

```

1000 ;*****
1001 ;*      *
1002 ;* CONTROLADOR *
1003 ;*      D E      *
1004 ;*      SPRITES  *
1005 ;*      *
1006 ;*****
1007 ;
1008      ORG 51000
1009 ;
1010 ; DETECCION DE CHOQUE
1011 ;
1012 CHOKE LD IX,(DEFADD)
1013      LD A,(IX+4)
1014      LD C,A
1015      CALL MAKHL
1016      CALL CHECK
1017      PUSH HL
1018      POP IX
1019      LD A,(IX+2)
1020      RLCA
1021      RLCA
1022      RLCA
1023      LD D,A
1024      ADD A,(IX+0)
1025      LD H,A
1026      LD A,(IX+3)
1027      RLCA
1028      RLCA
1029      RLCA
1030      LD E,A
1031      ADD A,(IX+1)
1032      LD L,A
1033      LD A,(NOSPR)
1034      AND A
1035      JP Z,RANGERR
1036      LD IX,INDEX
1037      LD B,A
1038      LD A,C
1039      EX AF,AF
1040      LD C,1
1041 MARCH
1042      LD A,(IX+2)
1043      AND A
1044      JR NZ,OK
1045 LECH
1046      INC IX
1047      INC IX
1048      INC IX
1049      INC IX
1050      INC IX
1051      INC IX
1052      INC C
1053      JR MARCH
1054 OK
1055      RLCA
1056      RLCA
1057      RLCA
1058      ADD A,(IX+0)
1059      SUB H

```

```

1060      CP D
1061      JR C,OKX
1062      JR Z,OKX
1063      NEG
1064      CP D
1065      JR NC,SIG
1066 OKX LD A,(IX+3)
1067      RLCA
1068      RLCA
1069      RLCA
1070      ADD A,(IX+1)
1071      SUB L
1072      CP E
1073      JR C,DADO
1074      JR Z,DADO
1075      NEG
1076      CP E
1077      JR C,DADO
1078 SIG
1079      DJNZ LECH
1080      LD BC,0
1081      RET
1082 DADO EX AF,AF
1083      CP C
1084      JR NZ,MUERT
1085      EX AF,AF
1086      JR SIG
1087 MUERT LD B,0
1088      RET
1089 ;
1090 ; CAMBIA EL COLOR
1091 ;
1092 CAMBC LD IX,(DEFADD)
1093      LD A,(IX+4)
1094      CALL MAKHL
1095      CALL CHECK
1096      LD B,(IX+12)
1097      PUSH HL
1098      POP IX
1099      PUSH BC
1100      CALL SPRITE
1101      POP BC
1102      LD (IX+4),B
1103      CALL SPRITE
1104      RET
1105 ;
1106 ; INICIALIZANDO EL SISTEMA
1107 ;
1108 INIT XOR A
1109      LD (NOSPR),A
1110      LD HL,INDEX
1111      LD DE,INDEX+1
1112      LD BC,245
1113      LD (HL),0
1114      LDIR
1115      RET
1116 ;
1117 ; BORRANDO DEFINICION
1118 ; Y SPRITE
1119 ;
1120 BORRA LD IX,(DEFADD)
1121      LD A,(IX+4)
1122      CALL MAKHL
1123      CALL CHECK
1124      PUSH HL
1125      POP IX
1126      LD A,(23693)
1127      LD (IX+4),A
1128      CALL SPRITE
1129      LD (IX+2),0
1130      LD HL,NOSPR

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

1131	DEC	(HL)		1202	PUSH	AF
1132	RET			1203	PUSH	DE
1133	CHECK	AND	A	1204	PUSH	BC
1134	JP	Z, RANGERR		1205	LD	A, (23693)
1135	CP	37		1206	LD	(IX+4), A
1136	JP	NC, RANGERR		1207	CALL	SPRITE
1137	LD	D, H		1208	POP	BC
1138	LD	E, L		1209	POP	DE
1139	INC	DE		1210	POP	AF
1140	INC	DE		1211	LD	(IX+4), A
1141	LD	A, (DE)		1212	LD	(IX+0), B
1142	AND	A		1213	LD	(IX+1), C
1143	RET	NZ		1214	LD	(IX+5), E
1144	LD	HL, MSG		1215	LD	(IX+6), D
1145	ERROR	PUSH	HL	1216	CALL	SPRITE
1146	LD	A, 1		1217	RET	
1147	CALL	#1601		1218	;	
1148	LD	B, 19		1219	;	DEFINIENDO UN SPRITE
1149	POP	HL		1220	;	
1150	ERRP	LD	A, (HL)	1221	DEFSPR	LD IX, (DEFADD)
1151	RST	16		1222	LD	A, (IX+4)
1152	INC	HL		1223	CALL	MAKHL
1153	DJNZ	ERRP		1224	LD	D, H
1154	LD	BC, (23621)		1225	LD	E, L
1155	CALL	#1A1B		1226	INC	DE
1156	LD	A, 58		1227	INC	DE
1157	RST	16		1228	LD	A, (DE)
1158	LD	C, (IY+13)		1229	AND	A
1159	LD	B, 0		1230	JR	Z, CREA
1160	CALL	#1A1B		1231	LD	HL, MSG1
1161	WAIT	BIT	5, (IY+1)	1232	JP	ERROR
1162	JR	Z, WAIT		1233	CREA	PUSH HL
1163	RST	8		1234	LD	A, (IX+12)
1164	DEFB	#FF		1235	LD	(HL), A
1165	MSG	DEFM 'Sprite no def	inido '	1236	LD	B, A
1166	MSG1	DEFM 'Sprite ya def	inido '	1237	LD	A, (IX+20)
1167	NOSPR	DEFB 0		1238	INC	HL
1168	;			1239	LD	(HL), A
1169	;	MOVIENDO UN SPRITE		1240	LD	C, A
1170	;			1241	CP	192
1171	MOVE	LD	IX, (DEFADD)	1242	JR	NC, RANGERR
1172	LD	A, (IX+4)		1243	INC	HL
1173	CALL	MAKHL		1244	LD	A, (IX+28)
1174	CALL	CHECK		1245	LD	(HL), A
1175	PUSH	HL		1246	RLCA	
1176	LD	A, (IX+12)		1247	RLCA	
1177	CP	192		1248	RLCA	
1178	JP	NC, RANGERR		1249	ADD	A, B
1179	LD	B, A		1250	CP	192
1180	INC	HL		1251	JR	NC, RANGERR
1181	INC	HL		1252	INC	HL
1182	LD	A, (HL)		1253	LD	A, (IX+36)
1183	RLCA			1254	LD	(HL), A
1184	RLCA			1255	RLCA	
1185	RLCA			C1256	RLCA	
1186	ADD	A, B		1257	RLCA	
1187	CP	192		1258	ADD	A, C
1188	JP	NC, RANGERR		1259	JR	C, RANGERR
1189	LD	A, (IX+20)		1260	INC	HL
1190	LD	C, A		1261	LD	A, (IX+44)
1191	INC	HL		1262	LD	(HL), A
1192	LD	A, (HL)		1263	LD	A, (IX+52)
1193	RLCA			1264	INC	HL
1194	RLCA			1265	LD	(HL), A
1195	RLCA			1266	LD	A, (IX+53)
1196	ADD	A, C		1267	INC	HL
1197	JP	C, RANGERR		1268	LD	(HL), A
1198	LD	D, (IX+29)		1269	POP	IX
1199	LD	E, (IX+28)		1270	CALL	SPRITE
1200	POP	IX		1271	LD	HL, NOSPR
1201	LD	A, (IX+4)		1272	INC	(HL)

1273	RET				1344	DJNZ	BUCLE
1274	RANGERR	RST	8		1345	EX	AF, AF
1275	DEFB	10	OUT OF RAN	GE.	1346	AND	A
1276	;				1347	POP	BC
1277	;	PONE UN SPRITE			1348	INC	B
1278	;				1349	JR	EXCES
1279	SPRITE LD	A, (IX+4)			1350	EXCES1 EX	DE, HL
1280	EX	AF, AF			1351	LD	H, O
1281	LD	B, (IX+2)			1352	LD	L, B
1282	LD	C, (IX+3)			1353	ADD	HL, HL
1283	LD	H, (IX+0)			1354	ADD	HL, HL
1284	LD	L, (IX+1)			1355	ADD	HL, HL
1285	LD	D, C			1356	ADD	HL, HL
1286	LD	E, H			1357	ADD	HL, HL
1287	PUSH	DE			1358	ADD	HL, DE
1288	LD	A, L			1359	DEC	L
1289	AND	7			1360	POF	BC
1290	PUSH	AF			1361	EXCES	LD A, C
1291	SUB	7			1362	AND	7
1292	NEG				1363	RET	Z
1293	LD	D, A			1364	EX	AF, AF
1294	ADD	A, A			1365	BUCLE2 LD	(HL), A
1295	ADD	A, D			1366	DEC	L
1296	LD	(REL+1), A			1367	DJNZ	BUCLE2
1297	CALL	DIR			1368	RET	
1298	LD	D, (IX+6)			1369	NEXTH	INC H
1299	LD	E, (IX+5)			1370	LD	A, H
1300	LD	A, C			1371	AND	7
1301	LD	C, B			1372	RET	NZ
1302	PON3				1373	LD	A, 32
1303	PUSH	AF			1374	ADD	A, L
1304	PUSH	HL			1375	LD	L, A
1305	PUSH	BC			1376	RET	C
1306	PON2				1377	LD	A, H
1307	PUSH	BC			1378	SUB	8
1308	PUSH	HL			1379	LD	H, A
1309	CALL	COLOR			1380	RET	
1310	EX	AF, AF			1381	SHIFT	LD C, O
1311	LD	(HL), A			1382	AND	A
1312	EX	AF, AF			1383	REL	JR SHIFT
1313	POP	HL			1384	RRA	
1314	LD	B, 8			1385	RR	C
1315	PON	LD A, (DE)			1386	RRA	
1316	CALL	SHIFT			1387	RR	C
1317	XOR	(HL)			1388	RRA	
1318	LD	(HL), A			1389	RR	C
1319	LD	A, C			1390	RRA	
1320	INC	L			1391	RR	C
1321	XOR	(HL)			1392	RRA	
1322	LD	(HL), A			1393	RR	C
1323	DEC	L			1394	RRA	
1324	CALL	NEXTH			1395	RR	C
1325	INC	DE			1396	RRA	
1326	DJNZ	PON			1397	RR	C
1327	POP	BC			1398	RET	
1328	DJNZ	PON2			1399	DIR	
1329	POP	BC			1400	LD	D, H
1330	POP	HL			1401	LD	E, L
1331	INC	L			1402	LD	A, D
1332	LD	B, C			1403	AND	%11000000
1333	POP	AF			1404	RRA	
1334	DEC	A			1405	RRA	
1335	JP	NZ, PON3			1406	RRA	
1336	LD	B, C			1407	OR	64
1337	LD	DE, 32			1408	LD	H, A
1338	CALL	COLOR			1409	LD	A, D
1339	POP	AF			1410	AND	%00111000
1340	JR	Z, EXCES1			1411	RLA	
1341	EX	AF, AF			1412	RLA	
1342	BUCLE	LD (HL), A			1413	LD	L, A
1343	ADD	HL, DE			1414	LD	A, D

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

1415      AND 7
1416      OR H
1417      LD H, A
1418      LD A, E
1419      SRL A
1420      SRL A
1421      SRL A
1422      ADD A, L
1423      LD L, A
1424      RET
1425 COLOR
1426      LD A, H
1427      AND #18
1428      SRA A
1429      SRA A
1430      SRA A
1431      ADD A, #58
1432      LD H, A
1433      RET
1434 MAKHL LD HL, INDEX
1435      LD D, 0
1436      LD E, A
1437      EX AF, AF
1438      DEC E
1439      LD A, E
1440      RLCA
1441      RLCA
1442      RLCA
1443      SUB E
1444      LD E, A
1445      ADD HL, DE
1446      EX AF, AF
1447      RET
1448 INDEX
1449 DEFADD EQU 23583

```

El programa es muy sencillo de utilizar y podremos hacerlo desde el Basic. Para ello sólo necesitamos, como ya se ha dicho, incluir la línea número 1 del programa número 2 en todos los programas donde queramos utilizar Sprites y, por supuesto, cargar la rutina en código máquina en la memoria del ordenador.

Las funciones que podemos realizar, todas desde el BASIC, son las siguientes:

- Inicializar el sistema.
- Definir un Sprite.
- Mover un Sprite de una posición de la pantalla a otra.
- Dar colores a los Sprites.
- Borrar un Sprite.
- Detectar si un Sprite ha chocado con otro.

Para realizar todas estas funciones utilizaremos funciones definidas (las de la línea 1) de la forma FN ver (parámetros). Conviene asignar el resultado de la función a una variable que no utilizemos en el programa, ya que si utilizamos dichas funciones con RANDOM-

ZE el resultado de los números aleatorios producidos por RND será muy monótono, y como los Sprites se suelen utilizar en juegos, éstos nos saldrán poco aleatorios.

Las funciones, tal y como están definidas en la línea número 1, son:

FN I() = Inicializa el sistema. Al principio del programa habrá que poner una instrucción parecida a LET ZZZZ = FN I() para poder utilizar los Sprites. Obsérvese que la variable utilizada ZZZZ no tiene aspecto de que se vaya a utilizar para otra función en todo el programa. No es necesario, ni acepta, ningún parámetro.

FN D (No. Sprite, Y, X, H, A, C, D) = Nos permite definir un Sprite. El significado de los parámetros es el siguiente:

No. Sprite = Número de Sprite que estamos definiendo. Ha de ser un número entre 1 y 36. Si el Sprite ya ha sido definido, el sistema nos dará un error.

X = Coordenada en X en alta resolución donde queremos que aparezca el Sprite.

Y = Coordenada en Y en alta resolución donde queremos que aparezca el Sprite.

H = Altura del Sprite medida en caracteres.

A = Anchura del Sprite medida en caracteres.

C = Color del Sprite al colocarlo en la pantalla. El número que pongamos ha de tener el siguiente formato:

$$C = \text{PAPER} + 8 * \text{INK} + 64 * \text{BRIGHT} + 128 * \text{FLASH}$$

si, por ejemplo, queremos un color en brillo, amarillo, con papel azul oscuro y sin flash, tendremos que poner:

$$C = 1 + 6 * 8 + 64 * 1 + 128 * 0 = 113$$

D = Dirección de memoria donde está definido el Sprite. La forma de definirlo es la siguiente: primero el primer carácter, luego, si hay más de una columna, el que esté debajo del primero, y así sucesivamente hasta completar la primera columna. Entonces empezamos con la segunda columna hasta terminarla. Esto se repite hasta que terminamos todas las filas. La figura 1 nos muestra el orden de definición de un Sprite de 3 x caracteres.

FN M (No. Sprite, Y, X, D) = Nos permite mover un Sprite desde su posición actual a una nueva. No. Sprite es el número de Sprite, X e Y son las coordenadas de la pantalla don-

de queremos posicionarlo. D es la dirección donde se encuentra definido el Sprite.

FN A (No. Sprite, C) = Nos sirve para cambiar el color de un Sprite. No. Sprite es el número de Sprite y C es el nuevo color que queremos que tenga.

FN B (No. Sprite) = Nos borra la definición del Sprite.

FN C (No. Sprite) = Nos permite saber si hemos chocado con otro Sprite. En caso negativo nos devuelve un cero (0). En caso afirmativo nos devuelve el número de Sprite con el que hemos chocado. Si chocamos con dos o más a la vez, nos devuelve el número más bajo de todos.

Es muy importante saber que, debido a que el programa utiliza la operación XOR para borrar un Sprite de su posición en la pantalla antes de colocarlo en una nueva posición, cuando chocan dos Sprites se produce una cierta confusión gráfica en la zona de pantalla donde esto ocurre. Una buena solución para

eliminar esta molestia es borrar los dos Sprites en conflicto.

La rutina, aparte de imprimir el mensaje de "Error out of range" cuando al definir o mover un Sprite nos salimos de la pantalla, añade al sistema operativo dos nuevos errores: "Sprite no definido" y "Sprite ya definido" cuando se intenta hacer algo con un Sprite que no existe o cuando ya está definido e intentamos redefinirlo sin haberlo borrado anteriormente. Por lo demás, los mensajes de error salen en el formato de siempre, indicando número de líneas y sentencia dentro de la línea donde se ha producido el error.

Ya para terminar, decir que al llamar a las sentencias FN, podemos incluir en la expresión cualquier tipo de fórmula que dé un resultado esperado. Por ello, una línea como la siguiente siempre será aceptada.

```
LET ZZZZ = FN D (cont, y + cy, x =  
cx, 2, 2, VAL P$, USR "A")
```


TRUCOS Y RUTINAS BASICAS



ONTINUANDO con los programas de representación de funciones en dos dimensiones para todos los ordenadores, vamos a ver a continuación un programa que nos permitirá representar cualquier función por com-

plicada que sea en nuestro IBM pc y compatible.

El programa ha sido desarrollado bajo GWBASIC para así poder ser utilizado por todos los compatibles del mercado que tengan este BASIC.

```
1000 REM *****
1010 REM *
1020 REM *  FUNCIONES GRAFICAS  *
1030 REM * ----- *
1040 REM *
1050 REM *****
1060 REM
1070 REM *****
1080 REM *
1090 REM * (c) Ediciones Siglo Cultural *
1100 REM *
1110 REM * (c) 1987
1120 REM *
1130 REM *****
1140 REM
1150 SCREEN 2
1160 REM
1170 REM *****
1180 REM *** INICIALIZACION ***
1190 REM *****
1200 REM
1210 OPTION BASE 1
1220 DIM A(3)
1230 LET A1=1
1240 LET A2=1
1250 LET X5=0
1260 LET AU=1
1270 DIM AU(2)
1280 DIM K1$(7)
1290 LET K1$(1)="T"
1300 LET K1$(2)="S"
1310 LET K1$(3)="S"
1320 LET K1$(4)="S"
1330 LET K1$(5)="S"
1340 LET K1$(6)="N"
1350 LET K1$(7)="N"
1360 KEY OFF
```

TRUCOS Y RUTINAS BASICAS

```

1370 FOR I=1 TO 10
1380   KEY I, ""
1390 NEXT I
1400 KEY 1, CHR$(13)+"GOTO 5700"+CHR$(13)
1410 REM
1420 REM *****
1430 REM *** PROGRAMA PRINCIPAL "MENU" ***
1440 REM *****
1450 REM
1460 VIEW
1470 CLS
1480 LOCATE 2,32
1490 PRINT "G R A F I C A S"
1500 PRINT TAB(32); "===== "
1510 PRINT:PRINT:PRINT
1520 DEF FN Y(X)=X/SIN(X)
1530 PRINT:PRINT:PRINT:PRINT:PRINT
1540 PRINT SPC(29); "1...Dibujar grafica."
1550 PRINT SPC(29); "2...Escribir formula."
1560 PRINT SPC(29); "3...Definir formato."
1570 PRINT SPC(29); "4...Salir del programa."
1580 PRINT:PRINT:PRINT
1590 PRINT:PRINT:PRINT
1600 LOCATE 24,31
1610 PRINT " <<ELIGE OPCION>>";
1620 LET K$=INKEY$
1630 IF K$="" THEN GOTO 1620
1640 LET K=ASC(K$)-48
1650 IF K<1 OR K>4 THEN GOTO 1620
1660 ON K GOSUB 2070,1980,1710,1680
1670 GOTO 1460
1680 CLS
1690 END
1700 REM
1710 REM *****
1720 REM *** SELECCION DEL FORMATO ***
1730 REM *****
1740 REM
1750 CLS
1760 PRINT
1770 PRINT " D E F I N I C I O N   D E L   F O R M A T O "
1780 PRINT "===== "
1790 PRINT
1800 PRINT
1810 RESTORE
1820 FOR F=1 TO 7
1830   READ AF$
1840   READ NE$
1850   READ ME$
1860   PRINT ME$;
1870   IF F=6 THEN PRINT "Margen cuando XY ==> infi. ( ";
1880   PRINT K1$(F); " )"; CHR$(29); CHR$(29);
1890   LET K$=INKEY$
1900   IF K$="" THEN GOTO 1890
1910   IF ASC(K$)>90 THEN LET K$=CHR$(ASC(K$)-32)
1920   IF K$<>" " THEN IF K$=AF$ OR K$=NE$ THEN LET K1$(F)=K$
1930   IF K$<>AF$ AND K$<>NE$ THEN PRINT ELSE PRINT K1$(F)
1940 NEXT F
1950 RETURN
1960 REM
1970 REM *****
1980 REM *** CAMBIO DE LA FUNCION ***
1990 REM *****
2000 REM
2010 CLS
2020 LOCATE 22,21

```

```

2030 PRINT "Cambiar la ecuacion y pulsar F1";
2040 LOCATE 10,20
2050 EDIT 1520
2060 GOTO 1460
2070 CLS
2080 REM
2090 REM *****
2100 REM *** DIBUJA EJES ***
2110 REM *****
2120 REM
2130 REM *** DIBUJO DEL RECUADRO ***
2140 REM
2150 VIEW(240,0)-(639,199)
2160 WINDOW(-200,-200)-(200,200)
2170 LINE(-200,0)-(-200,200),1
2180 LINE(-200,200)-(200,200),1
2190 LINE(200,200)-(200,-200),1
2200 LINE(-200,0)-(-200,-200),1
2210 LINE(-200,-200)-(200,-200),1
2220 REM
2230 REM *** DIBUJO DEL EJE X ***
2240 REM
2250 IF K1$(2)="N" THEN GOTO 2370
2260 LINE(5,38)-(-5,38),1
2270 LINE(5,76)-(-5,76),1
2280 LINE(5,114)-(-5,114),1
2290 LINE(200,0)-(-200,0),1
2300 LINE(5,152)-(-5,152),1
2310 LINE(5,190)-(-5,190),1
2320 LINE(5,-38)-(-5,-38),1
2330 LINE(5,-76)-(-5,-76),1
2340 LINE(5,-114)-(-5,-114),1
2350 LINE(5,-152)-(-5,-152),1
2360 LINE(5,-190)-(-5,-190),1
2370 PSET(0,0)
2380 REM
2390 REM *** DIBUJO DEL EJE Y ***
2400 REM
2410 IF K1$(3)="N" THEN GOTO 2530
2420 LINE(0,200)-(0,-200),1
2430 LINE(38,5)-(38,-5),1
2440 LINE(76,5)-(76,-5),1
2450 LINE(114,5)-(114,-5),1
2460 LINE(152,5)-(152,-5),1
2470 LINE(190,5)-(190,-5),1
2480 LINE(-38,5)-(-38,-5),1
2490 LINE(-76,5)-(-76,-5),1
2500 LINE(-114,5)-(-114,-5),1
2510 LINE(-152,5)-(-152,-5),1
2520 LINE(-190,5)-(-190,-5),1
2530 IF K1$(7)="S" THEN GOSUB 5390
2540 REM
2550 REM *****
2560 REM *** SELECCION DE ESCALA Y AUMENTO ***
2570 REM *****
2580 REM
2590 LOCATE 1,1
2600 PRINT "Escala del eje X:";A1
2610 LOCATE 1,19
2620 INPUT "",A1$
2630 IF A1$<>" " THEN LET A1=VAL(A1$)
2640 LOCATE 1,19
2650 PRINT A1;" "
2660 LOCATE 2,1
2670 PRINT "Escala del eje Y:";A2
2680 LOCATE 2,19
2690 INPUT "",A2$

```

TRUCOS Y RUTINAS BASICAS

```
2700 IF A2$<>"" THEN LET A2=VAL(A2$)
2710 LOCATE 2,19
2720 PRINT A2;" "
2730 LOCATE 12,31
2740 PRINT X=";INT(1000*200/A1)/1000
2750 LOCATE 1,58
2760 PRINT "Y=";INT(1000*200/A2)/1000
2770 LOCATE 3,1
2780 LET AU(1)=0
2790 PRINT "Ampliacion en eje X (0-9)";AU(1)
2800 LOCATE 3,27
2810 INPUT "",AU$(1)
2820 LOCATE 3,27
2830 PRINT AU$(1)
2840 IF AU$(1)<>"" AND AU$(1)<>"0" THEN GOTO 2850 ELSE GOTO 2900
2850 IF AU$(1)<>"1" AND AU$(1)<>"2" THEN GOTO 2880 ELSE GOTO 2900
2860 IF AU$(1)<>"3" AND AU$(1)<>"4" THEN GOTO 2870 ELSE GOTO 2900
2870 IF AU$(1)<>"5" AND AU$(1)<>"6" THEN GOTO 2880 ELSE GOTO 2900
2880 IF AU$(1)<>"7" AND AU$(1)<>"8" THEN GOTO 2890 ELSE GOTO 2900
2890 IF AU$(1)<>"9" THEN GOTO 2770
2900 LOCATE 4,1
2910 LET AU(2)=0
2920 PRINT "Ampliacion en eje Y (0-9)";AU(2)
2930 LOCATE 4,27
2940 INPUT "",AU$(2)
2950 LOCATE 4,27
2960 PRINT AU$(2)
2970 IF AU$(1)<>"" AND AU$(1)<>"0" THEN GOTO 2980 ELSE GOTO 3030
2980 IF AU$(1)<>"1" AND AU$(1)<>"2" THEN GOTO 2990 ELSE GOTO 3030
2990 IF AU$(1)<>"3" AND AU$(1)<>"4" THEN GOTO 3000 ELSE GOTO 3030
3000 IF AU$(1)<>"5" AND AU$(1)<>"6" THEN GOTO 3010 ELSE GOTO 3030
3010 IF AU$(1)<>"7" AND AU$(1)<>"8" THEN GOTO 3020 ELSE GOTO 3030
3020 IF AU$(1)<>"9" THEN GOTO 2900
3030 IF AU$(1)="0" THEN AU(1)=1
3040 IF AU$(1)="1" THEN AU(1)=10
3050 IF AU$(1)="2" THEN AU(1)=20
3060 IF AU$(1)="3" THEN AU(1)=30
3070 IF AU$(1)="4" THEN AU(1)=40
3080 IF AU$(1)="5" THEN AU(1)=50
3090 IF AU$(1)="6" THEN AU(1)=60
3100 IF AU$(1)="7" THEN AU(1)=70
3110 IF AU$(1)="8" THEN AU(1)=80
3120 IF AU$(1)="9" THEN AU(1)=90
3130 IF AU$(1)="0" THEN AU(1)=100
3140 IF AU$(2)="0" THEN AU(2)=1
3150 IF AU$(2)="1" THEN AU(2)=10
3160 IF AU$(2)="2" THEN AU(2)=20
3170 IF AU$(2)="3" THEN AU(2)=30
3180 IF AU$(2)="4" THEN AU(2)=40
3190 IF AU$(2)="5" THEN AU(2)=50
3200 IF AU$(2)="6" THEN AU(2)=60
3210 IF AU$(2)="7" THEN AU(2)=70
3220 IF AU$(2)="8" THEN AU(2)=80
3230 IF AU$(2)="9" THEN AU(2)=90
3240 IF AU$(2)="0" THEN AU(2)=100
3250 IF K1$(4)="S" THEN GOTO 3350
3260 REM
3270 REM *****
3280 REM *** SELECCION DE PASO (STEP) ***
3290 REM *****
3300 REM
3310 LOCATE 5,1
3320 PRINT "STEP:";S
3330 LOCATE 5,7
3340 INPUT "",S
3350 IF K1$(4)="S" THEN LET S=2/A1:LOCATE 5,1:PRINT "STEP: Autom.";S
```

```

3380 IF S=0 THEN LET S=2/A1
3370 IF K1$(1)="T" THEN GOTO 4430
3380 GOTO 3410
3390 END
3400 REM
3410 REM *****
3420 REM *** TRATAMIENTO GRAFICO DE LAS FUNCIONES ***
3430 REM *****
3440 REM
3450 REM *****
3460 REM *** DIBUJAR CON PUNTOS ***
3470 REM *****
3480 REM
3490 REM *** DIBUJO DE F(X) ***
3500 REM
3510 ON ERROR GOTO 5340
3520 LOCATE 8,1
3530 PRINT "Funcion: f(x)"
3540 PRESET(X5,0)
3550 LET XPOS=X5
3560 LET YPOS=0
3570 FOR X=X5 TO 200/A1 STEP S
3580   IF K1$(5)="S" THEN LOCATE 6,1:PRINT "X: ";XPOS; "      ":LOCATE 7,1:PRINT
      "Y: ";YPOS; "      "
3590   IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 3670
3600   IF K1$(6)="N" THEN IF XPOS>205 OR XPOS<-205 OR YPOS>205 OR YPOS<-205 THE
N GOTO 3710
3610   LET PX=INT(X*A1*AU(1))
3620   LET XPOS=PX
3630   LET PY=0
3640   LET PY=INT(FN Y(X)*A2*AU(2))
3650   LET YPOS=PY
3660   PSET(PX,PY)
3670 NEXT X
3680 REM
3690 REM *** DIBUJO DE F(-X) ***
3700 REM
3710 LOCATE 8,1
3720 PRINT "funcion: f(-x)"
3730 PRINT SPACE$(28)
3740 PRESET(-200/A1,0)
3750 LET XPOS=-200/A1
3760 LET YPOS=0
3770 FOR X=-200/A1 TO X5 STEP S
3780   IF K1$(5)="S" THEN LOCATE 6,1:PRINT "X: ";XPOS; "      ":LOCATE 7,1:PRINT
      "Y: ";YPOS; "      "
3790   IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 3870
3800   IF K1$(6)="N" THEN IF XPOS>205 OR XPOS<-205 OR YPOS>205 OR YPOS<-205 THE
N GOTO 3910
3810   LET PX=INT(X*A1*AU(1))
3820   LET XPOS=PX
3830   LET PY=0
3840   LET PY=INT(FN Y(X)*A2*AU(2))
3850   LET YPOS=PY
3860   PSET(PX,PY)
3870 NEXT X
3880 REM
3890 REM *** DIBUJO DE -F(X) ***
3900 REM
3910 LOCATE 8,1
3920 PRINT "Funcion: -f(x)"
3930 PRINT SPACE$(28)
3940 PRESET(X5,0)
3950 LET XPOS=X5
3960 LET YPOS=0
3970 FOR X=X5 TO 200/A1 STEP S

```

TRUCOS Y RUTINAS BASICAS

```

3980 IF K1$(5)="5" THEN LOCATE 8,1:PRINT "X:";XPOS;" "":LOCATE 7,1:PRINT
    "Y:";YPOS;" "
3990 IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 4070
4000 IF K1$(6)="N" THEN IF XPOS>205 OR YPOS>205 OR XPOS<-205 OR YPOS<-205 THE
N GOTO 4110
4010 LET PX=INT(X*A1*AU(1))
4020 LET XPOS=PX
4030 LET PY=0
4040 LET PY=INT(-(FN Y(X)*A2*AU(2)))
4050 LET YPOS=PY
4060 PSET(PX,PY)
4070 NEXT X
4080 REM
4090 REM *** DIBUJO DE -F(-X) ***
4100 REM
4110 LOCATE 8,1
4120 PRINT "Funcion:-f(-x)"
4130 PRINT SPACE$(28)
4140 PRESET(-200/A1,0)
4150 LET XPOS=-200/A1
4160 LET YPOS=0
4170 FOR X=-200/A1 TO X5 STEP S
4180 IF K1$(5)="S" THEN LOCATE 6,1:PRINT "X:";XPOS;" "":LOCATE 7,1:PRINT
    "Y:";YPOS;" "
4190 IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 4270
4200 IF K1$(6)="N" THEN IF XPOS>205 OR YPOS>205 OR XPOS<-205 OR YPOS<-205 THE
N GOTO 4280
4210 LET PX=INT(X*A1*AU(1))
4220 LET XPOS=PX
4230 LET PY=0
4240 LET PY=INT(-(FN Y(X)*A2*AU(2)))
4250 LET YPOS=PY
4260 PSET(PX,PY)
4270 NEXT X
4280 LOCATE 12,1
4290 PRINT "*****"
4300 LOCATE 13,1
4310 PRINT "*PULSA UNA TECLA*"
4320 LOCATE 14,1
4330 PRINT "*****"
4340 LET A$=INPUT$(1)
4350 GOTO 1460
4360 REM
4370 REM *****
4380 REM *** DIBUJAR CON TRAZO ***
4390 REM *****
4400 REM
4410 REM *** DIBUJO DE F(X) ***
4420 REM
4430 ON ERROR GOTO 5340
4440 LOCATE 8,1
4450 PRINT "Funcion: f(x)"
4460 PRINT SPACE$(28)
4470 PRESET(X5,0)
4480 LET XPOS=X5
4490 LET YPOS=0
4500 FOR X=X5 TO 200/A1 STEP S
4510 IF K1$(5)="S" THEN LOCATE 6,1:PRINT "X:";XPOS;" "":LOCATE 7,1:PRINT
    "Y:";YPOS;" "
4520 IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 4600
4530 IF K1$(6)="N" THEN IF XPOS>205 OR XPOS<-205 OR YPOS>205 OR YPOS<-205 THE
N GOTO 4640
4540 LET PX=INT(X*A1*AU(1))
4550 LET XPOS=PX
4560 LET PY=0
4570 LET PY=INT(FN Y(X)*A2*AU(2))

```

```

4580 LET YPOS=PY
4590 LINE -(PX,PY),1
4600 NEXT X
4610 REM
4620 REM *** DIBUJO DE F(-X) ***
4630 REM
4640 LOCATE 8,1
4650 PRINT "funcion:f(-x)"
4660 PRINT SPACE$(28)
4670 PRESET(-200,0)
4680 LET XPOS=-200
4690 LET YPOS=0
4700 FOR X=-200/A1 TO X5 STEP S
4710 IF K1$(5)="S" THEN LOCATE 6,1:PRINT "X:";XPOS;" " "Y:";YPOS;" " " ":LOCATE 7,1:PRINT
4720 IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 4800
4730 IF K1$(6)="N" THEN IF XPOS>205 OR XPOS<-205 OR YPOS>205 OR YPOS<-205 THE
N GOTO 4810
4740 LET PX=INT(X*A1*AU(1))
4750 LET XPOS=PX
4760 LET PY=0
4770 LET PY=INT(FN Y(X)*A2*AU(2))
4780 LET YPOS=PY
4790 LINE -(PX,PY),1
4800 NEXT X
4810 LOCATE 8,1
4820 REM
4830 REM *** DIBUJO DE -F(X) ***
4840 REM
4850 PRINT "funcion:-f(x)"
4860 PRINT SPACE$(28)
4870 PRESET(X5,0)
4880 LET XPOS=X5
4890 LET YPOS=0
4900 FOR X=X5 TO 200/A1 STEP S
4910 IF K1$(5)="S" THEN LOCATE 6,1:PRINT "X:";XPOS;" " "Y:";YPOS;" " " ":LOCATE 7,1:PRINT
4920 IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 5000
4930 IF K1$(6)="N" THEN IF XPOS>205 OR XPOS<-205 OR YPOS>205 OR YPOS<-205 THE
N GOTO 5040
4940 LET PX=INT(X*A1*AU(1))
4950 LET XPOS=PX
4960 LET PY=0
4970 LET PY=INT(-(FN Y(X)*A2*AU(2)))
4980 LET YPOS=PY
4990 LINE -(PX,PY),1
5000 NEXT X
5010 REM
5020 REM *** DIBUJO DE -F(-X) ***
5030 REM
5040 LOCATE 8,1
5050 PRINT "funcion:-f(-x)"
5060 PRINT SPACE$(28)
5070 PRESET(-200,0)
5080 LET XPOS=-200
5090 LET YPOS=0
5100 FOR X=-200/A1 TO X5 STEP S
5110 IF K1$(5)="S" THEN LOCATE 6,1:PRINT "X:";XPOS;" " "Y:";YPOS;" " " ":LOCATE 7,1:PRINT
5120 IF X*A1>32750 OR Y*A2>32750 OR X*A1<-32750 OR Y*A2<-32750 THEN GOTO 5200
5130 IF K1$(6)="N" THEN IF XPOS>205 OR XPOS<-205 OR YPOS>205 OR YPOS<-205 THE
N GOTO 5210
5140 LET PX=INT(X*A1*AU(1))
5150 LET XPOS=PX
5160 LET PY=0
5170 LET PY=INT(-(FN Y(X)*A2*AU(2)))

```

TRUCOS Y RUTINAS BASICAS

```
5180 LET YPOS=PY
5190 LINE -(PX,PY),1
5200 NEXT X
5210 LOCATE 12,1
5220 PRINT "*****"
5230 LOCATE 13,1
5240 PRINT "*PULSA UNA TECLA*"
5250 LOCATE 14,1
5260 PRINT "*****"
5270 LET A$=INPUT$(1)
5280 GOTO 1460
5290 REM
5300 REM *****
5310 REM *** TRATAMIENTO DE ERRORES ***
5320 REM *****
5330 REM
5340 LOCATE 9,1
5350 PRINT "** ERROR. SOBREPASA RANGO **"
5360 RESUME NEXT
5370 END
5380 REM
5390 REM *****
5400 REM *** DUBUJA CUADRICULA ***
5410 REM *****
5420 REM
5430 LINE(200,38)-(-200,38),1
5440 LINE(200,76)-(-200,76),1
5450 LINE(200,114)-(-200,114),1
5460 LINE(200,152)-(-200,152),1
5470 LINE(200,190)-(-200,190),1
5480 LINE(200,-38)-(-200,-38),1
5490 LINE(200,-76)-(-200,-76),1
5500 LINE(200,-114)-(-200,-114),1
5510 LINE(200,-152)-(-200,-152),1
5520 LINE(200,-190)-(-200,-190),1
5530 LINE(38,200)-(38,-200),1
5540 LINE(76,200)-(76,-200),1
5550 LINE(114,200)-(114,-200),1
5560 LINE(152,200)-(152,-200),1
5570 LINE(190,200)-(190,-200),1
5580 LINE(-38,200)-(-38,-200),1
5590 LINE(-76,200)-(-76,-200),1
5600 LINE(-114,200)-(-114,-200),1
5610 LINE(-152,200)-(-152,-200),1
5620 LINE(-190,200)-(-190,-200),1
5630 RETURN
5640 REM
5650 REM *****
5660 REM *** LINEAS DATA ***
5670 REM *****
5680 REM
5690 DATA "P","T","1...Punto o Trazo. ("
5700 DATA "S","N","2...Dibuja eje x. ("
5710 DATA "S","N","3...Dibuja eje y. ("
5720 DATA "S","N","4...STEP Auto. ("
5730 DATA "S","N","5...Escribe valores de x e y. ("
5740 DATA "S","N","6..."
5750 DATA "S","N","7...Dibuja cuadrícula. ("
5760 REM
5770 REM *****
5780 REM *** FIN DE LA DEFINICION DE LA FUNCION ***
5790 REM *****
5800 REM
5810 VIEW
5820 CLS
5830 PRINT "Si has definido el formato, tendras que volver a hacerlo."
```

```

5840 PRINT
5850 PRINT "PULSA UNA TECLA"
5860 LET A$=INPUT$(1)
5870 RUN

```

Algunas de las características más importantes de este programa son las siguientes:

- Posibilidad de anular el eje X.
- Posibilidad de anular el eje Y.
- Posibilidad de cuadricular la pantalla.
- STEP automático o manual.
- Posibilidad de dibujar con puntos.
- Posibilidad de dibujar con líneas.
- Dibujo de asíntotas cuando sean necesarias.
- Visualización de coordenadas durante el dibujo.
- Posibilidad de elección de escala en X y en Y.
- Posibilidad de ampliar la imagen en X.
- Posibilidad de ampliar la imagen en Y.
- Posibilidad de cambiar la función sin salir del programa.

Para elegir, o anular, algunas de estas opciones tenemos que pulsar la opción 3 del programa principal. Con ello nos aparecerán en pantalla una serie de preguntas que tenemos que ir contestando. Con esta opción definimos el formato en que queremos ver la representación de la función.



Fig. 1.

Si queremos cambiar la función a representar, sólo tenemos que pulsar la opción 2 del

menú. Nos aparecerá el número de línea que tiene almacenado la función. Entonces podemos modificarla con tranquilidad. Una vez hecho, basta pulsar F1 para volver al programa. El único inconveniente es que si previamente hemos definido el formato de la gráfica, tras definir la nueva función, nuestra definición se borrará.

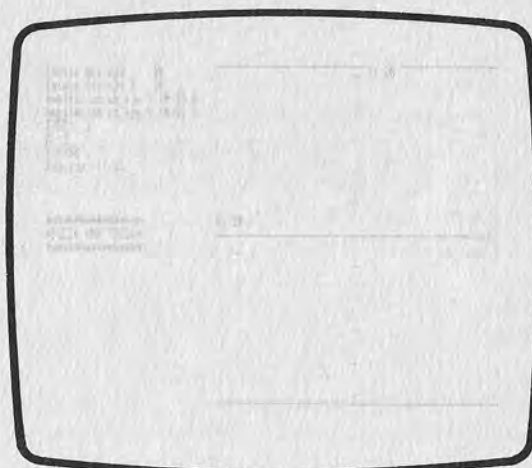


Fig. 2. $y = \text{sen}(x)/x$.

Una vez que hayamos pulsado la opción 1 para poder ver la representación de la función, el ordenador nos hará una serie de preguntas antes de ponerse a dibujar. Estas son:

— *Escala en X.* Inicialmente es 200, por lo que el número que introduzcamos dividirá a 200 y el resultado será la graduación del eje de las X. Así, si respondemos con un 10, la graduación será hasta 20, ya que $10 \times 20 = 200$.

— *Escala en Y.* Inicialmente también vale 200. El valor que introduzcamos dividirá este número y pondrá el resultado como máximo valor en el eje de las Y.

— *Ampliación en X.* Podemos hacer que la gráfica se amplíe en X para así poder ver mejor una parte de la misma. Se puede pulsar cualquier número entre 0 y 9.

— *Ampliación en Y.* Al igual que en X, podemos introducir un factor de ampliación entre 0 y 9.

TRUCOS Y RUTINAS BASICAS

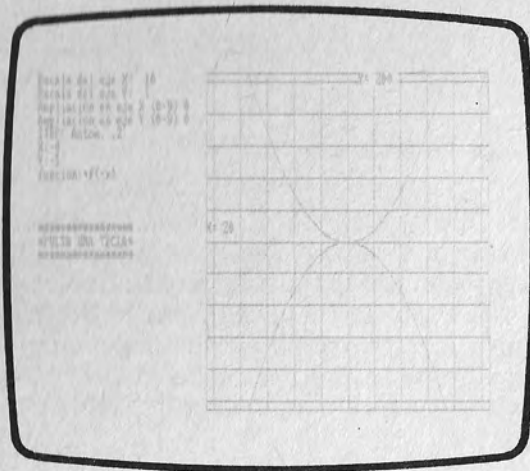


Fig. 3. $y = x^2$.

En el caso de que hubiésemos definido el STEP como manual, entonces el ordenador nos preguntará qué valor queremos que ten-

ga. Normalmente, si dejamos el STEP automático, el valor que el propio ordenador calculará será más que suficiente para nuestras necesidades.

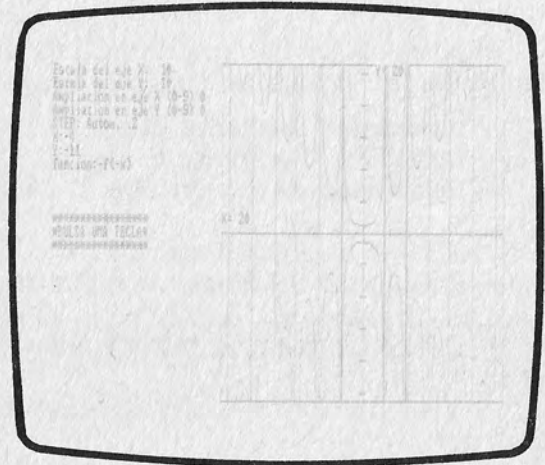
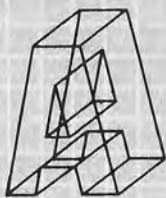


Fig. 4. $y = \text{sen}(x) \cdot x$.

EL TALLER DEL HARDWARE

Alimentación ininterrumpida



todo asiduo usuario de ordenador personal le habrá ocurrido alguna vez que después de introducir un programa con todo el cuidado del mundo o preparado un trabajo para su salida por impresora se ha ido todo al

cesto por una inoportuna interrupción de la alimentación. La red eléctrica tiene actualmente unas buenas características de estabilidad y continuidad, pero como nadie es perfecto puede ocurrir que se corte durante breves instantes la corriente debido a múltiples causas y que aunque la mayoría de los equipos alimentados con ella no se enteren, como son las bombillas o los electrodomésticos, sin embargo, ocasionen una pérdida total de la memoria de los ordenadores.

Las grandes instalaciones de ordenadores disponen de equipos de alimentación ininterrumpida para las máquinas que se consideran vitales, permitiendo con ello que puedan restaurarse las condiciones exactas del momento del apagón en un tiempo relativamente corto. El sistema de alimentación debe permitir salvar en disco o cinta todos los procesos que se encuentren en curso en el momento del fallo, de forma que al rearrancar automáticamente se recuperen las condiciones idénticas a las del instante en que se detectó el fallo. Esto suele requerir la carga en memoria de los procesos en el estado en que se encontraban, siendo de particular interés todos aquellos relacionados con operaciones en tiempo real o terminales de teleproceso, pues será necesario notificar a los usuarios remotos lo que ocurre para que tomen las medidas oportunas, aunque sea transitoriamente.

En general, el problema hay que resolverlo mediante dispositivos que suministren la energía que no puede recibirse de la red eléctrica. Cuando dicha energía es de pequeño volumen, los acumuladores o baterías pueden satisfacer la necesidad, mientras que si el consumo a suplir es muy grande será necesario disponer de grupos electrógenos que generen la corriente mediante máquinas rotativas. Cada vez es más fácil disponer de equipos que mediante circuitos electrónicos de potencia permitan generar la corriente eléctrica de forma similar a la de la red de distribución sin necesidad de máquinas rotativas, siendo todos los componentes de estado sólido por lo que se les suele denominar como alimentadores o inversores estáticos.

En el presente número vamos a describir equipos de alimentación ininterrumpida para ordenadores personales de pequeño consumo y para los de tipo intermedio.

Alimentación para máquinas tipo Spectrum

Los ordenadores que se alimentan en corriente continua y baja tensión son especialmente fáciles de alimentar en ausencia de corriente. Por lo general lo que pretendemos es facilitar que el contenido de la memoria pueda ser salvado en cinta o en diskette, pues si no hay alimentación general tampoco funcionará la pantalla o el televisor. Nos conformaremos con que el sistema trabaje unos minutos para permitir salvar el programa. También podemos suponer que el cassette en el que se salvan los datos o programas está alimentado por pilas.

Con una solución así de simple y económica también resolvemos un problema mu-

EL TALLER DEL HARDWARE

cho más molesto y habitual: son los microcortes. En determinados edificios o instalaciones que presentan deficiencias puede ocurrir que con frecuencia se produzcan bajadas de tensión momentáneas debido a la conexión de equipos de mucha potencia y que durante algunos ciclos reducen la tensión de alimentación a la mitad de la nominal. Si el número de ciclos durante los cuales se produce el fallo supera el que los condensadores de la entrada de la alimentación pueden mantener la

corriente consumida dentro de las especificaciones, se producirá una bajada brusca y una señal anómala de RESET del sistema que pueden dejar inutilizado un trabajo o nuestro juego favorito. A veces la pérdida de tensión no causa el borrado completo, pero la memoria, que al ser generalmente de tipo dinámico necesita refresco periódico, ha quedado borrada parcialmente, con lo que es necesario empezar de nuevo.

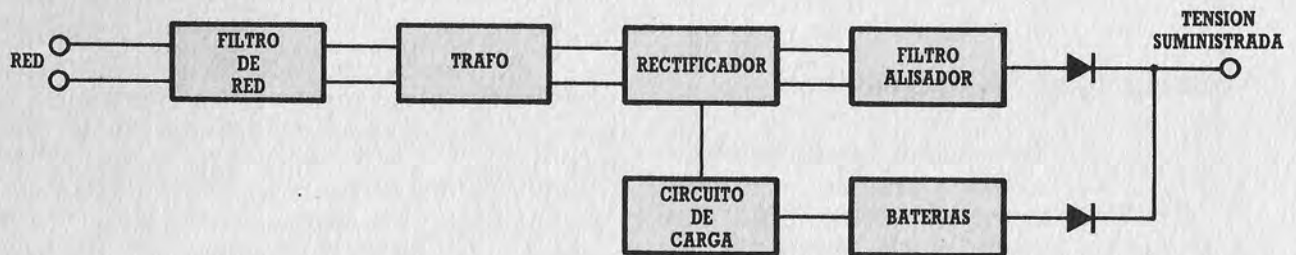


Fig. 1. Esquema de bloques.

El circuito de alimentación se compone de:

- Filtro de red, para reducir los impulsos transitorios generados en equipos próximos no desparasitados.
- Transformador de entrada, para reducir la tensión a la necesaria para el equipo.
- Rectificador.
- Filtro alisador de la señal rectificada.
- Diodos de bloqueo para la operación con baterías.
- Baterías.

— Circuito de carga de las baterías.

El equipo de alimentación ininterrumpida sustituirá al normalmente utilizado para alimentar el equipo y que suele venir con la máquina. En realidad, podría seguir utilizándose el alimentador normal, si se realiza exteriormente el conmutador para inserción de las baterías y la alimentación de éstas. Los ordenadores personales que se alimentan en corriente continua llevan internamente un regulador para suministrar las tensiones internas de forma estable.

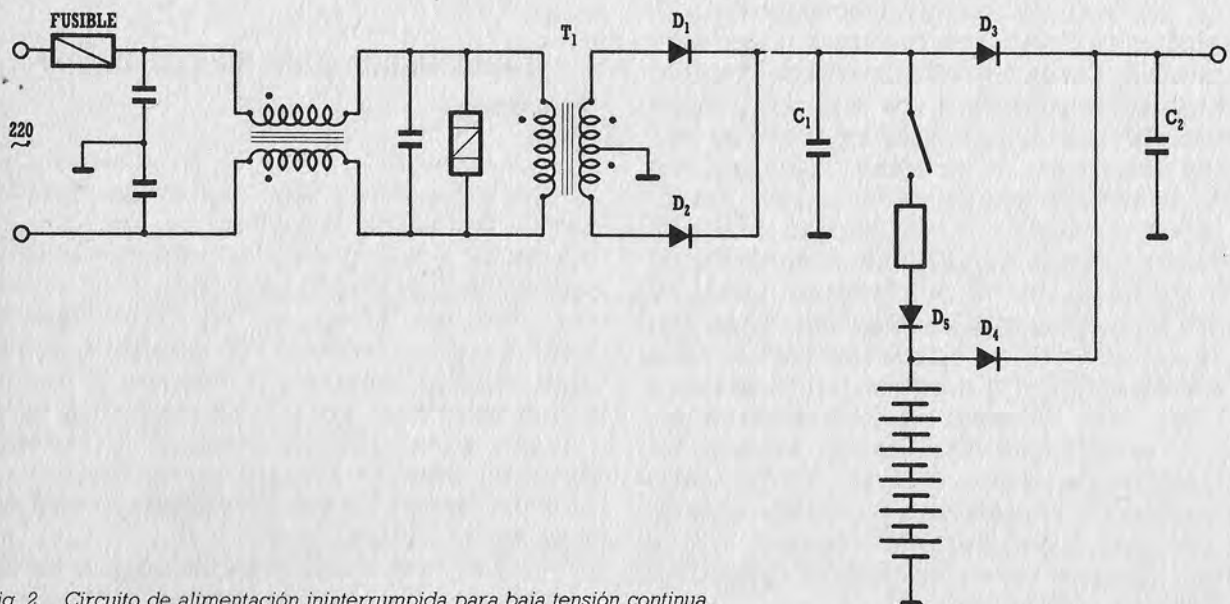


Fig. 2. Circuito de alimentación ininterrumpida para baja tensión continua.

El filtro de entrada es recomendable, pues con la combinación de condensadores y bobinas acopladas se eliminan los ruidos de alta frecuencia los microcortes, si son de muy corta duración. Los condensadores C a utilizar serán de unos pocos nanofaradios a 300 voltios. El varistor que se coloca a la salida anula los pulsos de muy elevada tensión que aparecen en la red al conmutarse equipos con elevada componente inductiva. El transformador se utilizará en modo reducción, para dar a la salida una tensión próxima a la nominal del ordenador, para que rectificada y filtrada presente un valor siempre por encima del mínimo especificado. Puede ser de toma media como el indicado o de salida simple, con el que sería necesario utilizar un puente de diodos como rectificador. La potencia nominal del transformador se seleccionará por lo menos un 25% por encima de la necesaria en el equipo.

El condensador C1 se dimensiona según la corriente a suministrar y el rizado admisible. Por ejemplo, para un rizado del 10% y una corriente de 1 amperio es necesario un condensador de unos 10.000 microfaradios.

Los diodos utilizados pueden ser del tipo 1N4001 o similares, para corrientes de hasta 1 amperio eficaz.

El circuito de carga de la batería se ha simplificado al máximo, reduciéndose a una resistencia limitadora de la corriente de carga y un diodo, para evitar la corriente en sentido contrario. La batería se cargará a un valor inferior a la tensión de salida normal, por lo que solamente entrará como alimentador cuando falle la corriente de entrada. El interruptor puede incluirse para permitir la carga a voluntad, aunque de forma automática cuando la batería se encuentre cargada completamente la corriente quedará reducida a un mínimo.

Los diodos de la salida impiden que en ningún momento alimenten de forma simultánea los dos circuitos, al ser siempre la tensión de la batería de valor inferior a la de la suministrada por la red cuando está en operación normal. En el momento del fallo de la alimentación el condensador de filtro se descargará, pasando el diodo inferior a conducir, suministrando toda la corriente la batería. Es necesario que la tensión suministrada sea superior al mínimo requerido por el ordenador, para que no se produzcan fallos al conmutar. Al volver la tensión de red tras un fallo, automáticamente se desconectará la alimentación de la batería, al superar la tensión en el condensador a la que entra desde ella.

Las baterías recargables son el elemento más costoso del diseño, siendo de un coste superior a diez veces a la de una pila desechable equivalente, por lo que en algunos casos podrían sustituirse por pilas de alta corriente, mucho más baratas, eliminando el circuito de carga.

Alimentación para equipos en corriente alterna

Muchos de los ordenadores personales tienen entrada directamente para la tensión normal de red de 220 voltios y 50 hertzios. Para alimentar a estos equipos de manera que no les afecten las caídas durante un determinado tiempo es necesario suministrarles una corriente de iguales características a las esperadas por los circuitos de entrada del equipo. Existen numerosas formas de conmutar la señal de alimentación, pero la más cómoda y práctica para poca potencia es la indicada en la figura, mediante un circuito de mando separado.



Fig. 3. Esquema de bloques.

EL TALLER DEL HARDWARE

La tensión de la red eléctrica es reducida el nivel de tensión interno adecuado y luego rectificada y filtrada. Esta tensión será de un valor similar al de las baterías utilizadas para el mantenimiento en los períodos de fallo de la alimentación. La tensión continua así generada es convertida en alterna mediante el circuito denominado inversor estático, por el hecho de «darle la vuelta» a la forma de onda. La tensión una vez convertida en alterna se aplica a los equipos de la misma forma que la tensión de red.

La conmutación entre la tensión rectificada de la red y la de las baterías se hace de igual forma que en el esquema anterior, mediante diodos de bloqueo. De los dos posibles circuitos que pueden alimentar solamente uno estará activo, es decir, suministrando corriente, pues el otro resultará bloqueado, al aparecer tensión en sentido inverso en el diodo. Para poder utilizar componentes económicos se suele reducir la tensión hasta los 12 voltios, que es la tensión nominal de las baterías de automóvil, resultando un equipo económico y de grandes prestaciones, aunque algo voluminoso. En equipos industriales que requieren más de una batería se conectan en serie para conseguir la tensión deseada.

Las características principales a considerar de la batería son: tensión nominal, tensión mínima de trabajo, carga máxima (expresada generalmente en amperios-hora), corriente máxima y vida útil. Según el consumo de nuestro equipo seleccionaremos una batería que pueda suministrar la corriente necesaria durante el tiempo máximo de fallo. Será necesario seleccionar un valor superior para tener un cierto margen de garantía.

Circuito inversor estático

Existen numerosos circuitos que permiten generar una tensión alterna de características similares a la de la red eléctrica, seleccionándose en cada caso según los requerimientos de potencia. Para las potencias requeridas en los ordenadores personales, del orden de 400 vatios como máximo, para una configuración casera típica, se suelen utilizar transistores de conmutación tanto de tipo bipolar, por su economía, como de tipo de efecto de campo (FET o MOS), por sus prestaciones. Los circuitos son muy similares en ambos casos. Para potencias superiores al kilowatio se em-

plean tiristores como elementos de conmutación, por presentar características idóneas tanto en corte como en conducción.

El equipo inversor se compone de dos bloques principales diferentes:

- Circuito de mando.
- Circuito de conmutación.

Para muy pequeñas potencias puede utilizarse una configuración de autoexcitación en la que el propio inversor forma parte del circuito oscilador de mando. Su gran ventaja radica en que la señal de excitación se consigue a través del transformador que también genera la potencia de salida, sin requerir circuito previamente alimentado para el control.

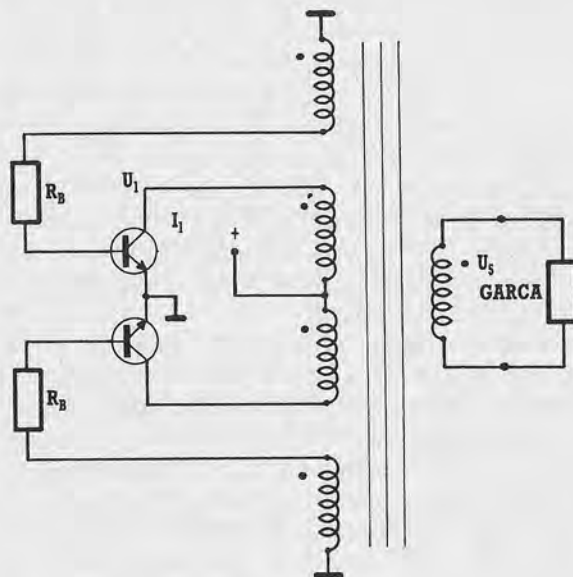


Fig. 4. Circuito inversor con autoexcitación.

El transformador se suele diseñar con núcleo magnético de ciclo de histéresis rectangular, para reducir en lo posible las pérdidas. Los bobinados superior e inferior constituyen la realimentación que mantiene las oscilaciones, pues por la conexión indicada hacen pasar alternativamente a cada transistor de conducción a corte. Los valores de las resistencias de carga han de ser un valor que permita saturar al transistor cuando deba entrar en conducción hasta el valor de corriente necesario.

La frecuencia de oscilación del equipo resulta determinada por las características magnéticas del núcleo y el valor de la carga,

por lo que sólo se usa cuando es necesario suministrar corriente alterna sin importar demasiado la frecuencia.

En aplicaciones generales es preferible

que la excitación se encuentre separada, para poder controlar independientemente los parámetros fundamentales: frecuencia, tensión, corriente de protección.

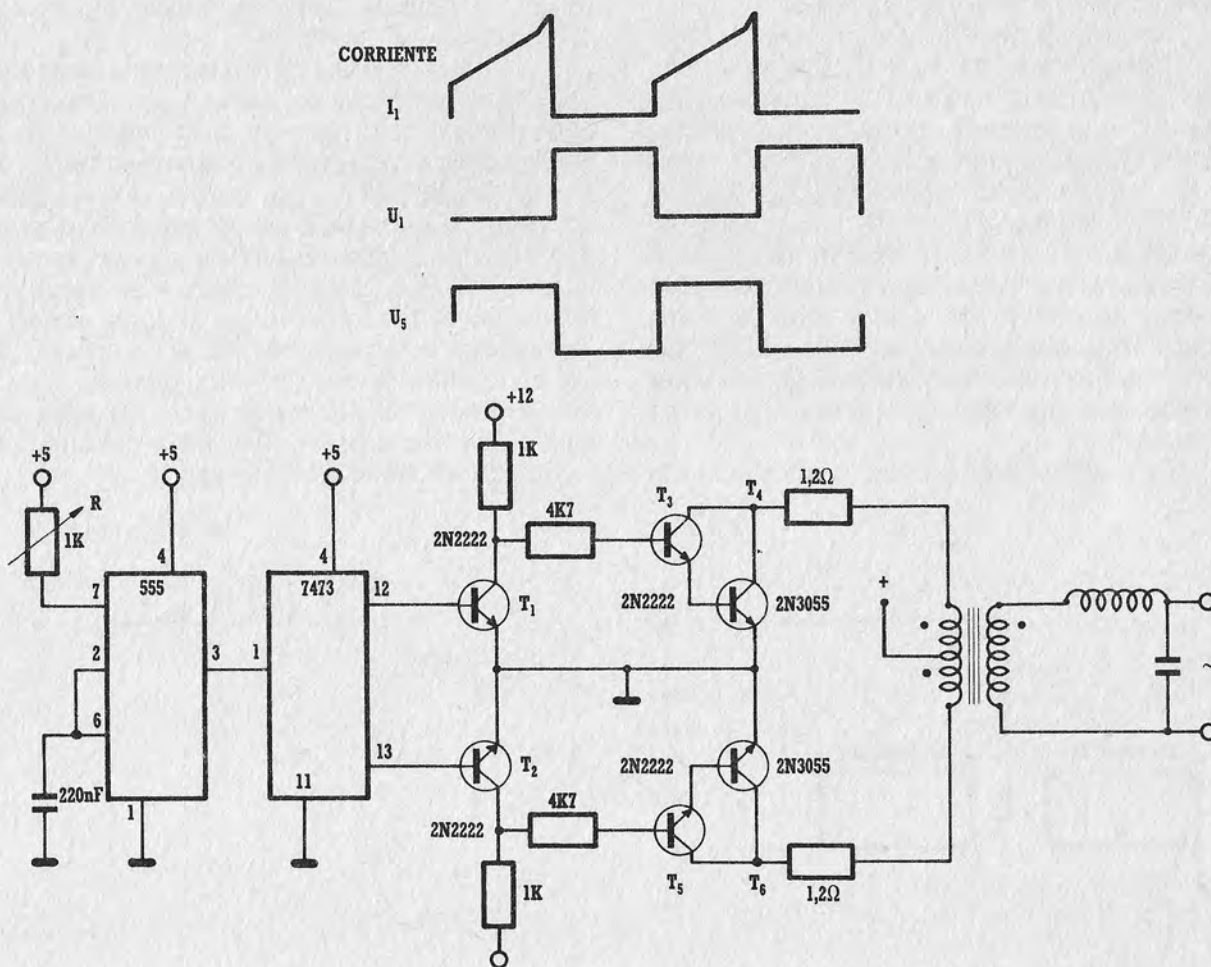


Fig. 5. Inversor con circuito de mando.

El circuito de mando incluye los componentes necesarios para la generación de la señal de referencia, la medida de las características de la tensión de salida y los circuitos de protección contra cortocircuitos y otros fallos de la carga alimentada. Un circuito de mando típico puede ser el de la figura compuesto por un oscilador con el circuito integrado LM555 y un biestable conectado como divisor de frecuencia. Mediante la resistencia variable R puede ajustarse la frecuencia de oscilación. A las entradas del oscilador se deberían conectar las señales indicadoras de sobrecorriente, para que se detenga la oscilación y por tanto la tensión en la salida, además de la alimentación de los transistores excitadores T1 y T2.

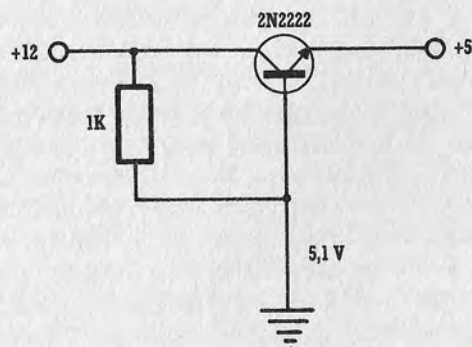


Fig. 6. Alimentación del circuito de mando.

La alimentación del circuito de mando puede hacerse desde la tensión regulada de salida, pero lo más normal es utilizar un pequeño transformador conectado directamente

EL TALLER DEL HARDWARE

a la entrada para que una vez rectificado suministra la energía necesaria en los circuitos de control y seguridad. En el esquema mostrado puede sacarse de la señal filtrada o de batería a través de un diodo Zener y un transistor que hace de regulador.

El circuito de potencia incluye los transistores conmutadores, T4 y T6, que están formados por un acoplamiento Darlington, para conseguir una elevada ganancia, el transformador de acoplamiento y el circuito filtro para que la onda de salida presente un contenido de armónicos dentro de los límites de los equipos a los que alimenta. El transformador utilizado puede ser uno normal de alimentación en conexión amplificación. Debe tenerse sumo cuidado al manipular la parte de salida del transformador conectado de esta forma, pues las tensiones que aparecen pueden resultar peligrosas.

Es conveniente colocar diodos en anti-

paralelo con los transistores de salida para evitar que las tensiones inversas que se producen en las conmutaciones puedan dañarlos, por superación de los valores máximos admisibles. Además permiten la circulación libre de la corriente de retorno, mejorando las características del inversor.

La forma de onda de la señal generada mediante conmutación de una señal continua será de tipo cuadrado, generalmente con un tiempo útil variable para permitir el control de la tensión de salida. Esta forma de onda tiene un elevado contenido en armónicos, que en muchas aplicaciones pueden causar trastornos, por lo que deberán eliminarse mediante filtros realizados con bobinas y condensadores. Al trabajar a frecuencias bajas el tamaño de los componentes resulta muy grande, por lo que es frecuente utilizar técnicas de conmutación a alta frecuencia con control del ciclo útil para simplificar el filtro de salida.

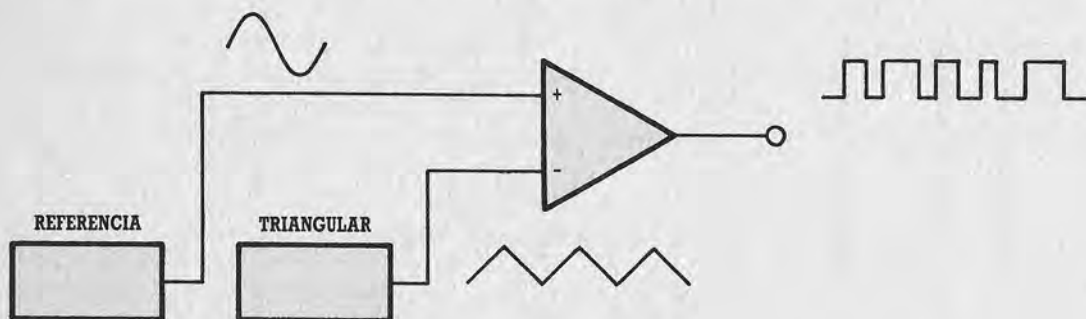


Fig. 7. Generación de forma de onda por conmutación.

El control del tiempo útil de cada ciclo para ajustarse a una forma de onda determinada se realiza mediante un circuito como el de la figura en la que un tren de pulsos triangulares de frecuencia fija se compara con la señal cuya forma se quiere sintetizar. En los períodos en los que la señal de referencia supera a la señal triangular se conmuta la tensión a la salida, anulándose en los períodos en los que la tensión de referencia es inferior. La señal resultante presenta la forma de la figura, constituida por pulsos de diferentes anchos. Al filtrar esta señal con un filtro de frecuencia de corte inferior a la frecuencia de repetición de pulsos del tren de conmutación, se eliminarán la mayor parte de los armónicos procedentes de la señal de pulsos de ancho

variable, resultando una señal de frecuencia como la de referencia con cierto rizado procedente del filtrado imperfecto.

El interés de utilizar señales de potencia de forma rectangular radica en que así la energía perdida en los elementos de control es mínima, pues o se encuentran en conducción total, con caída de tensión muy baja, o se encuentran fuera de conducción, con corriente despreciable. La potencia consumida en los elementos de conmutación es en realidad de pérdidas, por lo que interesa minimizarla todo lo posible. Los transistores de tipo MOS utilizados actualmente en los inversores estáticos cumple de forma casi ideal los requerimientos indicados, pues presentan una resistencia en conducción de décimas de ohmio, para

corrientes de decenas de amperios y resistencia muy elevada en el estado de no conducción. Por otra parte, los tiempos de conmutación de un estado a otro son muy reducidos, del orden de decenas de nanosegundo, por lo que las pérdidas son realmente muy bajas.

Otros componentes de importancia en circuitos conmutados de potencia son los diodos, pues si son de tipo normal de silicio presentan una caída de tensión, para corrientes de algún amperio que puede sobrepasar los 1,5 voltios, por lo que la potencia disipada puede ser alta. En las fuentes conmutadas de alta frecuencia y alto rendimiento se emplean diodos Schotky, que por su especial estructura de unión semiconductor-metal presentan una caída de 0,3 a 0,5 voltios, para intensidades similares a las indicadas.

■ Conclusión

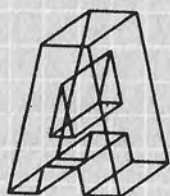
Con objeto de proteger nuestro ordenador personal contra el fallo de la alimentación, se han mostrado un par de circuitos para suministrar la energía necesaria en forma de corriente continua o alterna, según el tipo necesario. Debe recordarse la importancia de la elección de las baterías para que la duración del período esté dentro del rango de fallos posibles.

Tómense todas las precauciones posibles al generar tensiones alternas de valores tan altos como los de red, para evitar daños personales o deterioro de los equipos.

APRENDER CON EL ORDENADOR

NATURALEZA Y TECNOLOGIA

TIRO PARABOLICO



Lo igual que otros movimientos como el movimiento uniformemente acelerado, el movimiento circular, etc., el tiro parabólico es estudiado por la Cinemática. Estudia los disparos de los proyectiles sin salirse del campo gravitatorio terrestre, con una determinada inclinación de ángulo α , velocidad inicial v_0 , y aceleración constante. La trayectoria descrita por el proyectil es una parábola, siempre que tengamos en cuenta que no existe oposición del aire.

Componentes de la velocidad

Podemos conocer las componentes v_x y v_y de la velocidad del punto en función del tiempo. Primero descomponemos la velocidad v_0 en los ejes de coordenadas:

$$\begin{aligned}v_x &= v_0 \cos \alpha \\v_{y1} &= v_0 \sin \alpha\end{aligned}$$

La componente v_x no se modifica por no estar sometida a ninguna aceleración. La componente v_{y1} se ve afectada por el campo gravitatorio que produce una aceleración a lo largo del eje y , produciéndose sobre esta dirección un movimiento uniformemente acelerado.

La componente v_y será:

$$\begin{aligned}v_y &= v_{y1} + at. \quad (1) \\v_{y1} &= v_0 \sin \alpha \quad (2) \\a &= -g \quad (3)\end{aligned}$$

De (1), (2) y (3) se deduce que:

$$v_y = v_0 \sin \alpha - gt.$$

De acuerdo con lo anterior, las dos componentes de la velocidad son:

$$\begin{aligned}v_x &= v_0 \cos \alpha. \\v_y &= v_0 \sin \alpha - gt.\end{aligned}$$

Con estas dos ecuaciones podemos determinar la velocidad v en cualquier momento o instante:

$$v = \sqrt{v_x^2 + v_y^2}.$$

Coordenadas del punto móvil

La coordenada x del punto móvil es el espacio recorrido sobre el eje correspondiente con la velocidad v_x :

$$x = v_x t = v_0 t \cos \alpha.$$

La coordenada y es el espacio recorrido con movimiento uniformemente retardado sobre dicho eje. Por tanto:

$$\begin{aligned}y &= e_0 + v_{y1}t + \frac{1}{2}at^2. \\e_0 &= 0. \\v_{y1} &= v_0 \sin \alpha. \\a &= -g.\end{aligned}$$

Sustituyendo en las anteriores ecuaciones:

$$y = v_0 t \sin \alpha - \frac{1}{2}gt^2.$$

Tiempo que tarda en pasar por la horizontal

Cuando el punto móvil pasa por la horizontal, su ordenada es nula. Por tanto:

APRENDER CON EL ORDENADOR

$$y = v_0 t \sin \alpha - \frac{1}{2} g t^2$$

$$y = 0.$$

De esto se deduce:

$$v_0 t \sin \alpha - \frac{1}{2} g t^2 = 0 \Rightarrow t = 0 \text{ ó } t = \frac{2v_0 \sin \alpha}{g}$$

El resultado buscado es:

$$t = \frac{2v_0 \sin \alpha}{g}$$

■ Alcance máximo

El alcance máximo es la distancia máxima que alcanza el proyectil medida sobre la horizontal del punto desde el cual se ha hecho el disparo.

Sustituimos en la expresión de x el valor del tiempo que tarda el punto móvil en pasar por la horizontal, obteniendo así el valor de x en ese instante que es el alcance máximo:

$$x = v_0 t \cos \alpha$$

$$t = \frac{2v_0 \sin \alpha}{g}$$

El alcance máximo sería:

$$x_m = v_0 \frac{2v_0 \sin \alpha}{g} \cos \alpha = \frac{v_0^2 \sin 2\alpha}{g}$$

■ Tiempo que tarda en pasar por el punto máximo

En el punto máximo la componente vertical de la velocidad es nula. Debido a esto:

$$v_y = v_0 \sin \alpha - g t.$$

$$v_y = 0.$$

$$t = \frac{v_0 \sin \alpha}{g}$$

■ Altura máxima

La altura máxima es la máxima altura alcanzada por el disparo medida a partir del eje horizontal.

$$y = v_0 \sin \alpha - \frac{1}{2} g t^2$$

$$t = \frac{v_0 \sin \alpha}{g}$$

La altura máxima es:

$$y_m = \frac{v_0^2 \sin^2 \alpha}{2g}$$

A continuación se ofrece un ejemplo de ejecución del programa para unos valores concretos:



Fig. 1

```

10 REM *****
20 REM * PROGRAMA : TIRO PARABOLICO *
30 REM * VERSION : 1.0 *
40 REM * EQUIPOS : IBM PC Y COMPATIBLES *
50 REM *****
100 REM * PROGRAMA PARA EL CALCULO DE PARAMETROS DEL TIRO PARABOLICO *
110 PRINT"TIRO PARABOLICO"
120 PRINT"INTRODUZCA INCLINACION INICIAL DEL PROYECTIL EN RADIANES Y"
130 PRINT"LA VELOCIDAD EN METROS/SEGUNDO"
140 INPUT"INCLINACION ";IN
150 INPUT"VELOCIDAD INICIAL ";V0
160 LET G=9.8
165 GOSUB 2000
170 GOSUB 1090
180 GOSUB 1120
  
```

```

190 GOSUB 1150
200 GOSUB 1180
210 PRINT:PRINT:PRINT
220 PRINT"TIEMPO HASTA PASAR POR LA HORIZONTAL: ";FN R(TH);" SEGUNDOS"
230 PRINT"ALCANCE MAXIMO: ";FN R(XM);" METROS"
240 PRINT"INSTANTE DE PASO POR PUNTO MAXIMO: ";FN R(TP);" SEGUNDOS"
250 PRINT"ALTURA MAXIMA: ";FN R(YM);" METROS"
260 PRINT"¿DESEA ALGUN VALOR TEMPORAL DE LA VELOCIDAD (S/N)";
270 INPUT"";A$
280 IF A$="S" OR A$="s" THEN GOSUB 500
290 PRINT"¿DESEA REALIZAR OTRO TIRO (S/N)";
300 INPUT"";A$
310 IF A$="S" OR A$="s" THEN GOTO 110
320 GOTO 999
500 PRINT"INTRODUZCA EL INSTANTE DE MEDIDA";
510 INPUT"";T
520 GOSUB 1000
530 GOSUB 1050
540 PRINT"LA VELOCIDAD ES ";FN R(V);" M/S"
550 PRINT"LA COMPONENTE Vx DE LA VELOCIDAD ES ";FN R(VX);" M/S"
560 PRINT"LA COMPONENTE Vy DE LA VELOCIDAD ES ";FN R(VY);" M/S"
570 PRINT"LAS COORDENADAS DEL PUNTO MOVIL SON : "
580 PRINT"X(";T;")=";FN R(X);"METROS"
590 PRINT"Y(";T;")=";FN R(Y);"METROS"
600 PRINT"¿DESEA ALGUN VALOR TEMPORAL DE LA VELOCIDAD (S/N)";
610 INPUT"";A$
620 IF A$="S" OR A$="s" THEN GOTO 500
630 RETURN
999 END
1000 REM COMPONENTES DE LA VELOCIDAD
1010 LET VX=VO*COS(IN)
1020 LET VY=VO*SIN(IN)-G*T
1030 LET V=SGR(VX*VX+VY*VY)
1040 RETURN
1050 REM COORDENADAS DEL PUNTO MOVIL
1060 LET X=VX*T
1070 LET Y=VO*T*SIN(IN)-(G*T*T/2)
1080 RETURN
1090 REM TIEMPO QUE TARDA EN PASAR POR LA HORIZONTAL
1100 LET TH=2*VO*SIN(IN)/G
1110 RETURN
1120 REM ALCANCE MAXIMO
1130 LET XM=VO*VO*SIN(2*IN)/G
1140 RETURN
1150 REM TIEMPO DE PASO POR EL PUNTO MAXIMO
1160 LET TP=VO*SIN(IN)/G
1170 RETURN
1180 REM ALTURA MAXIMA
1190 LET YM=VO*VO*SIN(IN)*SIN(IN)/(2*G)
1200 RETURN
2000 DEF FN R(X)=INT(X*1000+.5)/1000
2010 RETURN

```

Modificaciones para otros equipos

SPECTRUM

LINEA 999 STOP

MATEMATICAS

■ Posición relativa de tres planos en el espacio

Sean las ecuaciones de los planos:

$$ax + by + cz + d = 0$$

$$ex + fy + gz + h = 0$$

$$jx + ky + lz + m = 0$$

A. Para que tengan puntos comunes ha de ser compatible el sistema, es decir:

$$r \begin{vmatrix} a & b & c \\ e & f & g \\ j & k & l \end{vmatrix} = r \begin{vmatrix} a & b & c & d \\ e & f & g & h \\ j & k & l & m \end{vmatrix} = r$$

a) Si $r = 3$, los tres planos se cortan en un punto único cuyas coordenadas se obtienen aplicando la regla de Cramer.

b) Si $r = 2$, una ecuación es combinación lineal de las otras dos y puede suprimirse. Los tres planos se cortan según una recta (son coaxiales).

c) Si $r = 1$, las tres ecuaciones son la misma; los planos son coincidentes.

B. Para que no tengan puntos comunes el sistema ha de ser incompatible, es decir:

$$r = r \begin{vmatrix} a & b & c \\ e & f & g \\ j & k & l \end{vmatrix} \neq r \begin{vmatrix} a & b & c & d \\ e & f & g & h \\ j & k & l & m \end{vmatrix} = r'$$

El rango de la primera matriz es menor o igual que el de la segunda, y los rangos de ambas matrices difieren como máximo en una unidad. Como los rangos son distintos las únicas posibilidades son:

APRENDER CON EL ORDENADOR

a) $r = 2, r' = 3$. Si en las dos primeras filas existe un menor principal se cortan según los dos primeros planos. Análogo razonamiento para los otros pares de filas.

b) $r = 1, r' = 2$. Los tres planos son paralelos.

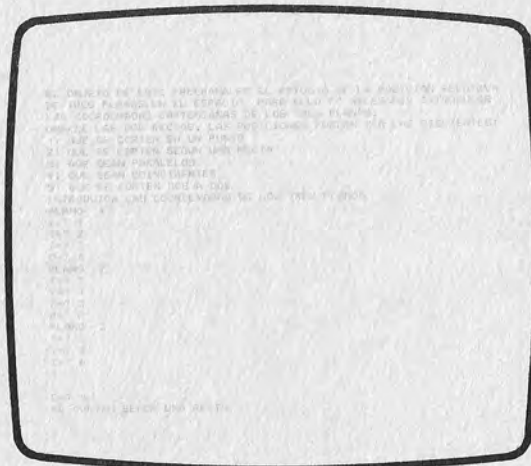


Fig. 2

```

10 REM *****
20 REM * PROGRAMA : POSICION RELATIVA DE TRES PLANOS EN EL ESPACIO *
30 REM * VERSION : 1.0 *
40 REM * EQUIPOS : IBM PC Y COMPATIBLES *
50 REM *****
100 CLS
110 PRINT"EL OBJETO DE ESTE PROGRAMA ES EL ESTUDIO DE LA POSICION RELATIVA"
120 PRINT"DE TRES PLANOS EN EL ESPACIO. PARA ELLO ES NECESARIO INTRODUCIR"
130 PRINT"LAS COORDENADAS CARTESIANAS DE LOS TRES PLANOS."
140 PRINT"UNA DE LAS DOS RECTAS. LAS POSICIONES PUEDEN SER LAS SIGUIENTES:"
150 PRINT"1) QUE SE CORTEN EN UN PUNTO"
160 PRINT"2) QUE SE CORTEN SEGUN UNA RECTA"
170 PRINT"3) QUE SEAN PARALELOS"
180 PRINT"4) QUE SEAN COINCIDENTES"
190 PRINT"5) QUE SE CORTEN DOS A DOS"
200 DIM P(4,3),N(3,3),C(3,3)
210 DEF FN T(X,Y)=C(X,Y)*C(X+1,Y+1)*C(X+2,Y+2)+C(X+2,Y+1)*C(X,Y+2)*C(X+1,Y)
      +C(X+1,Y+2)*C(X+2,Y)*C(X,Y+1)-(C(X+2,Y)*C(X+1,Y+1)*C(X,Y+2)
      +C(X,Y)*C(X+1,Y+2)*C(X+2,Y+1)+C(X+2,Y+2)*C(X,Y+1)*C(X+1,Y))
220 DEF FN D(X,Y)=C(X,Y)*C(X+1,Y+1)-C(X,Y+1)*C(X+1,Y)
300 PRINT"INTRODUZCA LAS COORDENADAS DE LOS TRES PLANOS"
310 FOR J=1 TO 3
320 PRINT "PLANO ";J;" "
330 INPUT"X=";P(1,J)
340 INPUT"Y=";P(2,J)
350 INPUT"Z=";P(3,J)
360 INPUT"D=";P(4,J)
370 NEXT J
380 GOTO 1000
400 REM * COMPROBACION DE RANGO 3 *
410 FOR I=1 TO 3
420 FOR J=1 TO 3
430 LET C(I,J)=N(I,J)
440 NEXT J
450 NEXT I
460 IF FN T(1,1)<>0 THEN LET R=3:GOTO 830
470 REM * COMPROBACION DE RANGO 2 *
480 LET R=0
490 FOR I=1 TO 2
500 FOR J=1 TO 2
510 IF FN D(I,J)<>0 THEN LET R=2
520 NEXT J
530 NEXT I
540 IF R=2 THEN GOTO 830
550 FOR J=1 TO 3
560 LET A=C(2,J)
570 LET C(2,J)=C(3,J)
580 LET C(3,J)=A
590 NEXT J
600 FOR J=1 TO 2
610 IF FN D(1,J)<>0 THEN LET R=2
620 NEXT J
630 IF R=2 THEN GOTO 830
640 FOR I=1 TO 2
650 LET A=C(I,2)
660 LET C(I,2)=C(I,3)
670 LET C(I,3)=A
680 NEXT I
690 IF FN D(1,1)<>0 THEN LET R=2:GOTO.830
700 FOR I=1 TO 3
710 FOR J=1 TO 3
720 LET C(I,J)=P(I,J)

```

```

730 NEXT J
740 NEXT I
750 FOR I=1 TO 3
760 LET A=C(I,2)
770 LET C(I,2)=C(I,2)
780 LET C(I,3)=A
790 NEXT I
800 FOR I=1 TO 2
810 IF FN D(I,1)<>0 THEN LET R=2
820 NEXT I
830 IF R=0 THEN LET R=1
840 RETURN
1000 REM *ROUTINA GENERAL*
1010 REM *RANGO DE LA MATRIZ A*
1020 FOR I=1 TO 3
1030 FOR J=1 TO 3
1040 LET N(I,J)=P(I,J)
1050 NEXT J
1060 NEXT I
1070 GOSUB 400
1080 IF R=3 THEN PRINT"SE CORTAN EN UN PUNTO":GOTO 9999
1090 LET RA=R
1100 REM * RANGO DE LA MATRIZ AMPLIADA *
1110 FOR I=2 TO 4
1120 FOR J=1 TO 3
1130 LET N(I-1,J)=P(I,J)
1140 NEXT J
1150 NEXT I
1160 GOSUB 400
1170 LET RB=R
1180 IF RB>RA THEN GOTO 1300
1190 FOR J=1 TO 3
1200 LET N(1,J)=P(1,J)
1210 NEXT J
1220 GOSUB 400
1230 IF RB<R THEN LET RB=R
1240 IF RB>RA THEN GOTO 1300
1250 FOR J=1 TO 3
1260 LET N(2,J)=P(2,J)
1270 NEXT J
1280 GOSUB 400
1290 LET RB=R
1300 REM * DETERMINACION DE POSICIONES *
1310 IF (RA=RB) AND (RA=2) THEN PRINT"SE CORTAN SEGUN UNA RECTA":GOTO 9999
1320 IF (RA=RB) AND (RA=1) THEN PRINT"SON COINCIDENTES":GOTO 9999
1330 IF (RA=2) AND (RB=3) THEN PRINT"SE CORTAN DOS A DOS":GOTO 9999
1340 IF (RA=1) AND (RB=2) THEN PRINT"SON PARALELOS":GOTO 9999
1350 PRINT"ERROR"
9999 END

```

Modificaciones para otros equipos

COMMODORE

LINEA 100 PRINT CHR\$(143)

SOCIEDAD

Adverbios

El adverbio es la parte de la oración que modifica al verbo, al adjetivo o a otro adverbio. Ejemplos: *Corrió* (verbo) *rápidamente* (adverbio). *Ayer estaba muy* (adverbio) *tranquila* (adjetivo). *Has hecho la tarea muy* (adverbio) *bien* (adverbio).

Según las circunstancias que expresen, los adverbios pueden ser:

De modo: Indican cómo se realiza la acción: Bien, despacio, lentamente, así, aprisa...

De tiempo: Cuándo se realiza la acción: ahora, nunca, tarde, mañana...

De lugar: Dónde se realiza la acción: aquí, allí, delante, arriba...

De cantidad: Cuánto alcanza la acción: mucho, nada, demasiado...

De afirmación: La acción se realiza: sí, también, ciertamente...

De negación: La acción no se realiza: no, tampoco, nunca, jamás...

De duda: No hay certeza de que la acción se realice: quizá (o quizás), acaso, tal vez...

```

10 REM *****
20 REM * PROGRAMA : ADVERBIOS *
30 REM * VERSION : 1.0 *
40 REM * EQUIPOS : IBM PC Y COMPATIBLES *
50 REM *****

```

APRENDER CON EL ORDENADOR

```
100 CLS
110 PRINT"EL SIGUIENTE TEST TIENE POR OBJETO COMPROBAR EL CONOCIMIENTO"
120 PRINT"DE LOS ADVERBIOS. EL ADVERBIO ES LA PARTE VARIABLE DE LA "
130 PRINT"ORACION QUE MODIFICA AL VERBO, AL ADJETIVO O A OTRO ADVERBIO"
140 PRINT"SE TRATA DE RESPONDER A QUE TIPO DE ADVERBIO PERTENECE EL QUE"
150 PRINT"SE PREGUNTE"
200 LET T=1
210 LET S=1
220 DIM A$(42,2)
230 FOR I=1 TO 7
240   READ X
250   LET T=S+X-1
260   READ T$
270   FOR J=S TO T
280     LET A$(J,1)=T$
290     READ A$(J,2)
300   NEXT J
310   LET S=S+X
320 NEXT I
330 LET R=0
400 FOR I=1 TO 10
410   LET A=INT(RND(1)*42)+1
420   PRINT"¿QUE TIPO DE ADVERBIO ES ";A$(A,2);
430   INPUT R$
440   IF A$(A,1)=R$ THEN PRINT"BIEN":LET R=R+1
450   IF A$(A,1)<>R$ THEN PRINT"MAL. LA RESPUESTA ERA ";A$(A,1)
460 NEXT I
470 PRINT"HAS OBTENIDO ";R;" ACIERTOS"
1000 DATA 10,MODO,BIEN,MAL,DEPRISA,DESPACIO,ASI,APRISA,DESPACIO,LENTAMENTE,
RAPIDAMENTE,DIRECTAMENTE
1010 DATA 8,TIEMPO,AHORA,NUNCA,TARDE,MAÑANA,AYER,DESPUES,ANTES,HOY
1020 DATA 8,LUGAR,AQUI,ALLA,ALLI,ACA,DELANTE,DETRAS,ARRIBA,DEBAJO
1030 DATA 5,CANTIDAD,MUCHO,POCO,NADA,TODO,DEMASIADO
1040 DATA 3,AFIRMACION,SI,TAMBIEN,CIERTAMENTE
1050 DATA 4,NEGACION,NO,TAMPOCO,NUNCA,JAMAS
1060 DATA 4,DUDA,GUIZA,GUIZAS,ACASO,TAL VEZ
```

Modificaciones para otros equipos

SPECTRUM

LINEA 220 DIM A\$ (42,2,10)

PARA LOS MAS JOVENES

■ Unidades de medida: longitud, masa, capacidad

El hombre a medida que pasaba el tiempo observó que tenía necesidad de estandarizar todo aquello que medía, pesaba y que tenía volumen, surgiendo así las unidades de medida (si, por ejemplo, tenía que medir el largo de una habitación mediante los pasos de dos personas que tuvieran distinta altura, aquella persona que fuera más alta mediría una longitud más pequeña que la persona que tuviera una altura menor, por lo que nunca se pondrían de acuerdo). Las principales unidades que surgieron fueron las de longitud, masa y volumen.

Unidades de longitud

Las unidades de longitud se utilizan para medir distancias (¿cuántos metros hay de mi casa al parque?).

La unidad principal en las medidas de longitud es el metro (unidad patrón). Existen unidades de longitud mayores y menores del metro con sus respectivas equivalencias.

Las unidades menores del metro son:

Decímetro (dm).

Centímetro (cm).

Milímetro (mm).

Las unidades mayores del metro son:

Kilómetro (Km).

Hectómetro (Hm).

Decámetro (Dm).

Como hemos dicho antes, existen equivalencias entre las distintas unidades de longitud:

1 m = 10 dm.

1 dm = 0,1 m.

1 m = 100 cm.

1 cm = 0,01 m.

1 m = 1000 mm.

1 mm = 0,001 m.

1 m = 1 m.

1 Km = 1.000 m.

1 Hm = 100 m.

1 Dm = 10 m.

Unidades de capacidad

Las unidades de capacidad son utilizadas para saber la cantidad de líquido que

cabe en una determinada medida. Al igual que sucedía en las medidas de longitud, existe la unidad principal, que en este caso es el litro (unidad patrón) con sus respectivas unidades mayores y menores.

Las unidades menores del litro son:

Decilitro (dl).

Centilitro (cl).

Mililitro (ml).

Las unidades mayores del litro son:

Kilolitro (Kl).

Hectolitro (Hl).

Decalitro (Dl).

Las equivalencias existentes son:

1 l = 10 dl.

1 dl = 0,1 l.

1 l = 100 cl.

1 cl = 0,01 l.

1 l = 1.000 ml.

1 ml = 0,001 l.

1 l = 1 l.

1 Kl = 1.000 l.

1 Hl = 100 l.

1 Dl = 10 l.

Unidades de masa

La unidad principal en las unidades de masa es el gramo (unidad patrón). Existen unidades mayores y menores del gramo:

Unidades menores del gramo:

Decigramo (dg).

Centigramo (cg).

Miligramo (mg).

Unidades mayores del gramo son:

Kilogramo (Kg).

Hectogramo (Hg).

Decagramo (Dg).

Las equivalencias existentes entre estas unidades son:

1 g = 10 dg.

1 dg = 0,1 g.

1 g = 100 cg.

1 cg = 0,01 g.

1 g = 1.000 mg.

1 mg = 0,001 g.

1 g = 1 g.

1 Kg = 1.000 g.

1 Hg = 100 g.

1 Dg = 10 g.

```

10 REM *****
20 REM * PROGRAMA : UNIDADES DE MEDIDA *
30 REM * VERSION : 1.0 *
40 REM * EQUIPOS : IBM PC Y COMPATIBLES *
50 REM *****
100 DIM P$(21),A$(21),E(21)
110 FOR I=1 TO 21
120   READ P$(I)
130   READ A$(I)
140   READ E(I)
150 NEXT I
160 CLS
200 CLS
210 PRINT"ELIGE OPCION:"
220 PRINT"1) TEST DE ABREVIATURAS DE LAS UNIDADES DE LONGITUD, CAPACIDAD Y MASA"
230 PRINT"2) TEST DE EQUIVALENCIAS DE LAS UNIDADES DE LONGITUD, CAPACIDAD
240 PRINT" Y MASA"
250 PRINT"ELIGE OPCION:";
260 INPUT "";O$
270 IF O$="1" THEN GOTO 300
280 IF O$="2" THEN GOTO 600
290 GOTO 200
300 REM * TEST DE ABREVIATURAS DE UNIDADES DE MEDIDA *
310 PRINT"*****"
320 PRINT"¿CONOCES LAS ABREVIATURAS DE LAS UNIDADES DE LONGITUD,"
330 PRINT"CAPACIDAD Y MASA?"
340 PRINT"SI CONSIDERAS QUE ES ASI, ATREVETE CON ESTE SENCILLO TEST."
350 PRINT" Tendrás que escribir la abreviatura de la unidad que se te"
360 PRINT"pregunte; recuerda que hay algunas que se escriben con"
370 PRINT"mayúsculas pero otras se escriben con minúsculas."
380 PRINT" Tendrás que responder a diez preguntas. Al final recibirás"
390 PRINT"la calificación de la prueba"
400 LET R=0
410 FOR I=1 TO 10
420   LET X=INT(RND(1)*21)+1
430   PRINT"¿CUAL ES LA ABREVIATURA DE ";P$(X);" ?";
440   INPUT"";R$
450   IF R$=A$(X) THEN PRINT "BIEN":LET R=R+1
460   IF R$<>A$(X) THEN PRINT "NO. LA RESPUESTA ES ";A$(X)
470 NEXT I
480 PRINT"HAS TENIDO ";R;" ACIERTOS"
490 GOTO 780
600 PRINT"*****"
610 PRINT"¿CONOCES LAS EQUIVALENCIAS DE LAS UNIDADES DE LONGITUD,"
620 PRINT"CAPACIDAD Y MASA?"

```

```

630 PRINT"SI CONSIDERAS QUE ES ASI, ATREVETE CON ESTE SENCILLO TEST."
640 PRINT"TENDRAS QUE ESCRIBIR LA EQUIVALENCIA DE LA UNIDAD QUE SE TE"
650 PRINT"PREGUNTE; RECUERDA QUE HAY ALGUNAS SON DE LA FORMA 0.01"
660 PRINT"ES DECIR, SON DE FORMA DECIMAL"
670 PRINT"TENDRAS QUE RESPONDER A DIEZ PREGUNTAS. AL FINAL RECIBIRAS"
680 PRINT"LA CALIFICACION DE LA PRUEBA"
690 LET R=0
700 FOR I=1 TO 10
710   LET X=INT(RND(1)*21)+1
720   PRINT"&CUANTOS ";P$(INT((X-1)/7)*7+4);"S ES UN ";P$(X);" ";
730   INPUT"";RE
740   IF RE=E(X) THEN PRINT "BIEN":LET R=R+1
750   IF RE<>E(X) THEN PRINT "NO. LA RESPUESTA ES ";E(X)
760   NEXT I
770 PRINT"HAS TENIDO ";R;" ACIERTOS"
780 END
1000 DATA KILOLITRO,Kl,1000,HECTOLITRO,Hl,100,DECALITRO,Dl,10,LITRO,l,1
1010 DATA DECILITRO,dl,0.1,CENTILITRO,cl,0.01,MILILITRO,ml,0.001
1020 DATA KILOMETRO,Km,1000,HECTOMETRO,Hm,100,DECAMETRO,Dm,10,METRO,m,1
1030 DATA DECIMETRO,dm,0.1,CENTIMETRO,cm,0.01,MILIMETRO,mm,0.001
1040 DATA KILOGRAMO,Kg,1000,HECTOGRAMO,Hg,100,DECAGRAMO,Dg,10,GRAMO,g,1
1050 DATA DECIGRAMO,dg,0.1,CENTIGRAMO,cg,0.01,MILIGRAMO,mg,0.001

```

Modificaciones para otros equipos

SPECTRUM

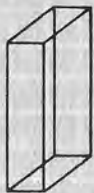
LINEA 100 DIM P\$ (21,10), A\$ (21,2), E(21)
 LINEA 780 STOP

COMMODORE

LINEA 200 PRINT CHR\$(143).

PEQUEÑA HISTORIA DE LA INFORMATICA

Los modelos 7044 y 7090 de IBM



BM era YA en los años sesenta una gran empresa multinacional, líder en el campo de la informática. Nada tiene, pues, de extraño que uno de sus principales clientes fuera el Ministerio de Defensa de los Estados Unidos de América, con el que merecía la pena esforzarse en el cumplimiento de los contratos. La realidad es que pagaban seguro y sin especiales complicaciones, y se les podía considerar un campo sin límites, ya que cualquier labor experimental, por cara que fuere, si alcanzaba algún objetivo podía ser vendida sin especiales problemas a alguno de los Organismos relacionados con dicho Ministerio.

En 1958 el Ministerio de Defensa de los Estados Unidos solicitó de IBM una máquina específica para el proceso de los datos obtenidos en el Sistema de Defensa Antimisiles BMEWS (Ballistic Missile Early Warning System). Aunque el contrato era muy interesante, IBM no se encontraba en una buena situación, ya que dada la importancia del uso que pensaba darse al ordenador, existían cláusulas con penalizaciones serias para el caso en el que no se cumplieran fechas o especificaciones. El ordenador debería ofrecer un nivel de prestaciones muy superior a los que presentaban los ordenadores que ofrecía IBM en aquel momento, y para ello, naturalmente, exigirían mucha experimentación y desarrollo, traduciéndose todo ello en tiempo, del que IBM andaba verdaderamente escaso en esa ocasión. En algo más de un año, IBM debía disponer de los equipos, con lo que el trabajo debería realizarse a contrarreloj, y, por

otro lado, la gran empresa tampoco podía comprometer su prestigio fallando.

El laboratorio de desarrollo aconsejó basar el nuevo producto en el 709, que parecía disponer de características aceptables, aunque, desde luego, sería necesario mejorarlo para conseguir que fuera más rápido, y posiblemente también para obtener unas precisiones mayores.

Para lograr una mayor velocidad, era importante reducir el ciclo de base a menos de dos microsegundos, y ello mediante procesadores con circuitos transistorizados cuyas características de velocidad no eran en absoluto fáciles de conseguir en esas fechas.

IBM sabía desde la firma del contrato que no podría cumplirlo, pero estaba decidida a aceptar el desafío, y confiaba en poder salir del atolladero de alguna forma. Efectivamente, llegado el momento, como la instalación de la máquina debía realizarse en los locales definitivos, y la puesta en marcha del sistema duraba algún tiempo, IBM envió sus máquinas en fecha, junto con un "ejército de especialistas" que irían puliendo poco a poco sus defectos, naturalmente, sin aceptarlos como debidos al propio ordenador, por principio. La puesta a punto de la máquina se había llevado a cabo en fábrica, y todos esos problemas "eran sólo los normales, debidos a adaptarla a las necesidades del usuario in situ".

Pero mientras la labor anterior se iba realizando, IBM se iba metiendo en más problemas. Por un lado, ¿debía ofrecer a sus clientes el nuevo producto, casi sin probar, o era mejor esperar a pulir sus deficiencias, evitándose infinidad de problemas? La empresa no podía olvidar su parque de clientes, casi todos de alquiler, a los que había ofrecido, ofrecía o debía ofrecer sus últimos modelos. Estos podían devolverle el antiguo 709 y pedirle el

PEQUEÑA HISTORIA DE LA INFORMATICA

nuevo TX, si aparecía en catálogo. Pero el mercado es como un enorme grupo de jugadores de mus. Si se descartan de las cartas malas, están dando al mismo tiempo otras oportunidades al contrario, de forma que siempre es importante saber cortar en el momento adecuado, aunque las cartas de que se disponga no sean las óptimas. Si IBM no incluía el 709TX en catálogo, y se tomaba un tiempo adicional, otras compañías que estaban investigando en productos similares le tomarían la delantera, y su magnífico contrato podía pasar a ser un mal negocio para la Compañía.

IBM supo acertar en esta decisión. Se arriesgó e incluyó la máquina en catálogo, vendiendo cerca del medio centenar, lo que suponía un éxito considerable, dado el alto precio de los ordenadores, que superaban los dos millones de dólares.

A pesar del éxito del 7090, y de la enorme potencia económica de las empresas norteamericanas, lo cierto es que el mercado, aunque suficientemente grande, podría ampliarse bastante más si las máquinas tuvieran un precio mucho más bajo. Naturalmente, lo que no podía hacer IBM es cobrar más al que más tiene, y menos al que menos puede. Pero sí podía partir del prototipo y recortar algunas de sus características para poder vender "otra" máquina, mucho más barata. La nueva máquina era el 7040, basada en el 7090, pero con ciertas restricciones, sobre todo en lo concerniente al manejo de direcciones, y registros índice. A pesar de sus pequeños inconvenientes, frente al 7090, dado el enorme abaratamiento y las innegables cualidades de la máquina, ésta tuvo una enorme aceptación.

La idea de aprovechar diseños de ordenadores anteriores, en los que se habría invertido mucho dinero para modificarlos para alguna tarea específica, como se ve, no es nueva en absoluto, pero, sin embargo, en la segunda generación va aplicándose cada vez más. Y así van apareciendo modelos pensados para uso científico, empresarial, militar, etc. Sin embargo, la segunda generación de ordenadores se caracteriza principalmente por la aparición de grandes sistemas centralizados, en contraposición a lo que hoy en día conocemos: grandes sistemas, redes, pequeños y medianos sistemas, microordenadores para multitud de tareas, etc. En los años cincuenta o sesenta, los informáticos eran generalmente matemáticos, personas que trabajaban en despachos «perdidos» de organismos estatales o paraestatales, de los que el grupo más importan-

te solía tener algo que ver con el Ejército. Al ir avanzando en el tiempo, el informático pasa a despachos de grandes compañías, pero su trabajo sigue siendo oscuro a los ojos de los demás. Ya no tiene por qué ser un matemático, pero sigue muy ligado al hardware de la máquina. Poco a poco, sin embargo, la informática se va desarrollando y perdiendo sus cualidades esotéricas para ir interesando más y más al público. Pero no nos perdamos, volvamos a la segunda generación de ordenadores, para citar a un ordenador muy especial, el PDP1, una de las mejores máquinas del momento, y además el primer ordenador que dispuso de pantalla gráfica. El PDP1 era un producto de Digital Equipment Corporation, firma creada en 1962 por los hermanos Olson y Harland Anderson. Ken Olson, uno de los fundadores, había trabajado en el MIT (Massachusetts Institute of Technology) en el proyecto Whirlwind, y estaba absolutamente convencido de que trabajando con seriedad, invirtiendo el tiempo necesario para que los productos salieran absolutamente probados, podrían ir poco a poco ganándose un prestigio en el mundo de los ordenadores. El PDP1 había sido diseñado por Olson para trabajar en tiempo real, y a su rapidez de proceso se unía un sistema muy eficaz de acceder a la memoria. El PDP1 era una máquina relativamente cara, comparada con lo que existía en el mercado, de forma que sólo fue vendida para tareas muy específicas de investigación, etc. A pesar de las innegables ventajas que presentaba la pantalla de gráficos, en uno de los modelos posteriores, el PDP4, ésta fue eliminada.

Otra máquina, basada en prototipos anteriores, pero pensada para objetivos distintos, fue el 1401 IBM. Este ordenador se había diseñado pensando específicamente en el mercado de las empresas, por entonces informatizadas a un nivel muy bajo (lectoras de fichas y otras máquinas de oficina) o incluso sin ninguna mecanización en absoluto. El ordenador base, del que había partido el 1401 era el 702, pero se había modernizado bastante, y se ofrecía a un precio bajísimo, con el fin de abrir mercado en el mundo empresarial. La IBM consiguió efectivamente introducirse, y vendió más de diez mil ordenadores de esta gama. Incluso sacó otros dos modelos posteriores, de mejores características (sobre todo más rápidos), el 1440 y el 1460.

Del 1401 se ofrecían varias versiones, algunas de las más simples con una unidad de entrada/salida consistente únicamente en una

lectora de fichas perforadas, y, por supuesto, reduciendo memoria y sistemas de almacenamiento al mínimo. Naturalmente, el usuario siempre podría mejorar el equipo, adquiriendo discos, o cintas magnéticas, otras unidades de entrada/salida más sofisticadas, etc.

Como ordenador pensado para uso científico, el más representativo de su época fue el CAB 500, ordenador francés muy original, fabricado por la compañía SEA.

Una de las características especiales de este ordenador es que no era demasiado voluminoso, ni requería condiciones de temperatura y humedad muy críticas, lo que lo hacía muy adaptable a cualquier entorno de oficina.

El CAB 500 era original en muchos aspectos. Disponía de dos sistemas de entrada de datos. Uno mediante lectora de fichas perforadas (la lectora era semejante a las de los teletipos) y otro simplemente a través del teclado. La salida se realizaba utilizando fichas o cinta perforada que eran leídas e impresas mediante un teletipo. El sistema de comunicación del usuario a través del teclado, hoy en día tan utilizado, supuso un enorme avance, hizo los ordenadores mucho más accesibles, tanto si el usuario era un experto en programación, como si no lo era tanto.

Otra de las características muy especiales del CAB 500 es que podía microprogramarse, es decir, el usuario podía modificar los microprogramas del ordenador. El hecho de que los microprogramas que controlan las funciones básicas del ordenador puedan ser modificables (y no sólo los programas de aplicación que sobre estos microprogramas funcionan normalmente) exige el diseño de una arquitectura del procesador diferente y sumamente original. Los microprogramas del CAB 500 estaban almacenados en los tambores magnéticos, junto con los restantes programas normales, de forma que unos y otros se podían utilizar sin problemas. El CAB 500 era un ordenador de vanguardia en el momento de su aparición. En efecto, resultaba muy flexible el que estuviera microprogramado y se pudiera a su vez microprogramar. De este ordenador se vendieron unas cincuenta máquinas, pero desgraciadamente, esta firma francesa desapareció del mercado poco después.

La microprogramación durante los años cincuenta era una característica bastante poco común. La razón fundamental para que así fuera era principalmente técnica. Como las microinstrucciones o instrucciones elementales

residían en la memoria del ordenador, y la tarea de la unidad de control se reduce en este caso a la transferencia de control de un registro a otro, el tiempo de acceso a la memoria debe ser relativamente corto, porque si no, para cualquier operación sencilla, se perdería un tiempo enorme en las lecturas. Por esta razón, hasta que no se comercializaron las memorias de toros de ferrita, cuyos tiempos de acceso mejoraron bastante, no era en absoluto útil disponer de la facilidad de microprogramación.

Las casas fabricantes comenzaron a preocuparse de la microprogramación por razones, como siempre, económicas. La compatibilidad entre sus máquinas era un problema constante. Por ello, muchas casas comenzaron a desarrollar unos programas "emuladores", o microprogramas, que "imitaban" el comportamiento de otra máquina cualquiera, realizando las operaciones en código máquina, de forma que ordenadores de distintas firmas pudieran trabajar con programas de otros ordenadores.

El Honeywell 800 podía microprogramarse perfectamente para emular el IBM 1401. De este modo, cuando la IBM se decidió a eliminar esa gama y pasar a la 360, la Honeywell ganó innumerables clientes para los que resultaba mucho más sencillo comprar un Honeywell 800 con su emulador que cambiar toda la documentación, o al menos parte de ella.

¿Sabía usted que...

Wiener fue el padre de la Cibernética?

En 1948 Norbert Wiener, uno de los científicos teóricos más importantes de nuestro siglo, publicó en París un libro titulado *Cybernetics or Control and Communication in the Animal and the Machine*. El libro, pese a haber aparecido en París, está escrito en inglés. Wiener era un intelectual muy sincero y honrado en sus concepciones. Era una persona muy especial, del que existen numerosísimas anécdotas, casi todas ellas relacionadas con su despiste, que podía llegar a niveles difícilmente creíbles. En muchas ocasiones, estando en clase, se le ocurría alguna idea original, y al seguirla en su pensamiento, llegaba a olvidarse de que los alumnos seguían esperando que continuara la clase que les estaba impartiendo. Un día

PEQUEÑA HISTORIA DE LA INFORMATICA

un amigo se lo encontró caminando por la calle en pijama. Wiener estaba dándole vueltas a un problema, y no sabía ni por qué calle circulaba. El padre de la Cibernética era una persona encantadora. A pesar de su enorme valía, no estaba en absoluto pagado de sí mismo. Tenía una bien ganada fama de honrado y leal, y no aceptaba honores ni prebendas como recompensa de su trabajo. Siendo profesor de Matemáticas del MIT (Massachusetts Institute of Technology), estaba muy interesado por el progreso de la técnica, y feliz de desarrollar su trabajo con sus alumnos, que disculpaban sus enormes despistes.

Durante la segunda guerra mundial, Wiener trabajó para la Armada de los Estados Unidos, como muchos otros matemáticos. Recordemos lo necesitado que estaba el Gobierno Americano de personal para el cálculo de trayectorias, y lo lenta que era la tarea. En su aproximación a las armas, Wiener comprendió que todos los avances y descubrimientos que se llevaran a cabo en la industria de armamento se acabarían por aplicar, tarde o temprano. Los políticos utilizan siempre a los científicos, y todos los avances técnicos en el campo militar serían terribles retrocesos para la Humanidad.

Wiener y Von Neumann trabajaron juntos un corto período de tiempo, pero eran dos personalidades totalmente distintas, y Wiener comprendió rápidamente que nunca llegaría a formar un equipo compacto con Von Neumann. Es cierto de Von Neumann era una persona afable y muy agradable, pero desde su llegada a los Estados Unidos estuvo siempre implicado en algún sentido con temas militares de defensa. En sus primeros años, se ocupó principalmente del desarrollo de las nuevas máquinas u ordenadores que realizaban los cálculos, y en los últimos años de su vida, estuvo relacionado muy estrechamente con el proyecto de desarrollo de la bomba atómica. Wiener era absolutamente distinto. Había reflexionado mucho sobre los últimos avances técnicos, que iban sucediéndose a velocidad vertiginosa y había llegado a dos convencimientos íntimos muy importantes: jamás estos nuevos descubrimientos debe-

rían utilizarse para dominar o destruir, y jamás se deberían diseñar armas, con fines en principio disuasorios, pues tarde o temprano éstas se utilizarían. Para controlar este último punto, Wiener consideraba muy útil la creación de unos comités de control de los que formaran parte científicos, con el fin de regular el uso dado a las armas que ellos habían ayudado a diseñar.

Wiener era un gran pensador. Desde la publicación de su libro *Cybernetics*, estaba muy ocupado escribiendo otro, que intentaba unificar temas dispersos, relacionándolos siempre con dos directrices maestras: el control y la comunicación. Su segundo libro, *Cybernetique et Societe*, fue tan revolucionario y nuevo como el primero, aunque estaba en cierto modo basado en él.

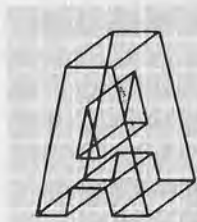
Wiener estaba interesado en estudiar el control de los procesos. Este control podía darse tanto en fenómenos naturales como artificiales, tanto en humanos y animales como en máquinas, e incluso en procesos sociales, mucho más amplios. A la ciencia que estudiaba estos campos, sin desarrollar por el momento, Wiener denominó Cibernética, palabra que deriva de la palabra griega cuyo significado es piloto. Partiendo de estas ideas sobre el control y el gobierno de los procesos, se podía pasar a estudiar el fenómeno de la "toma de decisiones". Esto último era válido tanto en la política como en la economía, o en otras ciencias sociales.

Poco después, con esta nueva concepción filosófica los grandes científicos del momento comenzaron a interesarse sobre los lenguajes, e incluso sobre el conocimiento.

Paralelamente a estos estudios de tipo filosófico, en los Estados Unidos, un neurofisiólogo llamado McCulloch utilizaría la Cibernética para la creación de modelos artificiales que copiarían los esquemas de razonamiento humano, en suma, el esquema cerebral del hombre.

Durante un tiempo, la Cibernética parecía no interesar a los padres del ordenador. En fechas actuales ha recobrado toda su vigencia.

■ Aplicaciones del proceso de imágenes en la industria



UNQUE hace ya bastante tiempo (más de 25 años) que se están procesando imágenes mediante ordenador, sus aplicaciones prácticas en la industria (no a nivel de laboratorio), no tienen ni mucho menos esa edad. La razón principal de este retraso se debe a que para que una aplicación cualquiera de proceso de imágenes sea aceptada por la industria, es necesario que aporte soluciones de una precisión suficiente, y a enorme velocidad (para su aplicación en tiempo real); y este nivel técnico sólo se obtenía (hace unos años) a unos costes ciertamente prohibitivos. Los métodos alternativos son mucho más conocidos, accesibles para el industrial que sin ser un especialista en informática deberá tomar la decisión. En efecto, el beneficio debe ser claro y directo, como, por ejemplo, una evidente reducción de los costes de producción, etc., o la aplicación de proceso de imágenes será rechazada siempre, o casi siempre, por razones económicas. También puede suceder que, aunque el beneficio no sea tan directo, el sistema mejore las condiciones de trabajo de los empleados, y, éstos, al trabajar más relajados, obtengan rendimientos más altos. Un tercer y último caso, no tan probable, es el empresario, aficionado a la informática, con deseos de estar mecanizado con tecnología punta, que quiera destacarse de los demás y dar algún otro pequeño servicio extra, aunque esto le cueste algún dinero más.

Efectivamente, la viabilidad económica de las aplicaciones de proceso de imágenes también ha estado en tela de juicio en otros estamentos distintos del industrial, aunque en

ellos, el capítulo económico tenía y tiene siempre menos importancia que en el industrial. En este último, la rentabilidad es su motor y su razón de ser. Un ejemplo fácil de cómo las instituciones estatales tienen una visión muy distinta lo tenemos en el campo de la medicina. La Seguridad Social comprará aparatos caros que resuelvan problemas médicos adecuadamente ayudando a salvar vidas, sin considerar tanto su precio. No debemos, sin embargo, olvidar otros campos en los que también estas aplicaciones se ven desde un prisma distinto al comercial, como, por ejemplo, el militar (reconocimientos aéreos), en cartografía, astronomía, química, etc., la televisión, que puede recurrir a muchas de estas aplicaciones. El reconocimiento de caracteres, la transmisión de imágenes, su grabación, y posterior análisis, son utilísimos en muchos campos. Es, pues, claro que el inconveniente mayor para adoptar estas aplicaciones suele ser su coste. Sin embargo, en la industria existen muchas tareas para las que el proceso de imágenes puede suponer beneficios tan obvios y evidentes que el número de industrias que han adoptado este tipo de aplicaciones desde su aparición ha ido aumentando casi sin pausa.

Siguiendo los pasos de la industria informática y electrónica, las aplicaciones de proceso de imágenes por ordenador han ido bajando de precio, de forma que su viabilidad ha ido creciendo con el tiempo. A ello ha contribuido el bajo coste de ordenadores cada vez más potentes, capaces de realizar acciones que hace unos años sólo máquinas de cierto volumen podían realizar.

Considerando el proceso de las imágenes desde el punto de vista de la industria, podríamos clasificarlas en dos grupos generales: uno de ellos en el que el proceso de las imágenes por ordenador viene a ser una especie de sustitución de la visión humana, y otro, en el que podría sustituir algunas tareas de tra-

TEMAS MONOGRAFICOS DE VANGUARDIA

tamiento de gráficos. Al primer grupo se suele designar como visión por ordenador, y algunas de las tareas que realiza han sido (o podrían haber sido) realizadas por una persona humana. Al segundo grupo, que está creciendo mucho en estos últimos años, podríamos referirnos como el proceso de fotografías, y otras imágenes o gráficos de tipo artístico. En el campo de la fotografía, por ejemplo, los métodos ópticos están siendo sustituidos por procesos realizados en un ordenador, sobre todo en ciertos entornos, como el militar, cartográfico, etc.

En lo que respecta a la visión artificial, o visión por ordenador, podríamos dividirla en cuatro grupos diferenciados: por un lado, los robots, o máquinas capaces de "ver", 2) tareas de inspección, 3) conversión de trazados con líneas, y finalmente 4) modelismo de objetos. De estos cuatro grupos, los dos primeros apartados tienen una enorme importancia. En todo caso, la razón de ser para cada uno de estos grupos es la sustitución de una tarea que implique visión humana, bien porque la tarea deba realizarse en condiciones peligrosas, o insanas, agotadoras, etc.

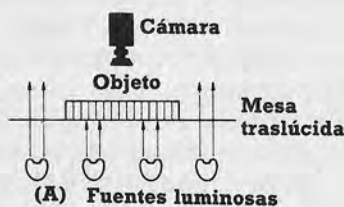
Al referirnos aquí a las aplicaciones industriales del proceso de imágenes, es útil recordar que existen diferencias sustanciales entre el campo industrial y otros campos. En la industria se aquilata cuidadosamente la inversión, y las imágenes obtenidas tienen siem-

pre el grado de bondad mínimo necesario para cada aplicación. Generalmente, las aplicaciones complejas requieren ordenadores costosos, y, aunque algunas pueden realizarse en ordenadores pequeños (PC y compatibles), el tiempo necesario en estos casos es muy superior, incidiendo, naturalmente, en el coste de la aplicación.

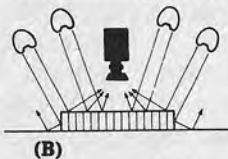
Además, para que la calidad de las imágenes sea buena, aceptablemente libre de ruidos e interferencias, se hace imprescindible controlar exquisitamente la posición de la cámara en el entorno, y sobre todo la iluminación. Esta última, si se estructura adecuadamente, puede simplificar enormemente la interpretación de las imágenes. En cuanto a la resolución, normalmente se utilizan valores de 256×256 ó 512×512 pixels, por comodidad de proceso en el ordenador.

El manejo de objetos o materiales mediante visión por ordenador

Es una de las aplicaciones de visión artificial más importantes en la industria. La tarea a realizar podría subdividirse en tres operaciones: a) identificación del objeto dentro de un conjunto de objetos sin clasificar, b) determinación de la posición del objeto dentro de



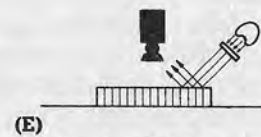
Iluminación posterior



Iluminación frontal difusa



Iluminación frontal



Iluminación frontal polarizada



Iluminación por luz estructurada

un espacio bi o tridimensional, y c) determinación de la orientación de objeto. Por otro lado, éste puede estar estático o en movimiento. Es fácil suponer que c) es el paso más difícil, ya que el objeto puede tomar en principio cualquier posición y orientación dentro del espacio que se considera, e incluso puede estar «tapado» por otros objetos que se encuentren delante de él.

No todas las aplicaciones requieren la misma «calidad» de imagen, ni siquiera deben cumplir las tres operaciones anteriores. Recordemos que en las aplicaciones industriales manda el dinero, y, si la tarea no es imprescindible, y es susceptible de ser eliminada, el coste de la aplicación puede reducirse considerablemente. Así, existen muchas aplicaciones en las que la orientación no es imprescindible, o incluso la posición del objeto (piénsese, por ejemplo, en una central lechera con una cadena de botellas de leche de las que se desea inspeccionar si están en buenas condiciones o tienen alguna lacra, con el peligro de que se rompan). Es evidente, que en este caso, la identificación del objeto no tiene sentido alguno, ya que todas son botellas de leche prácticamente idénticas. En otras cintas de transporte, los objetos pueden haber sido colocados por otra máquina o por personal de una manera precisa, de forma que no se requiera en absoluto averiguar la posición y orientación del objeto, que serán siempre fijas. El caso más sencillo se da cuando el objeto se deposita de forma estable sobre una superficie o cinta, y el más complicado, cuando se encuentra «revuelto» entre un montón de objetos de cualquier tipo, y, por tanto, su posición y orientación puede ser cualesquiera. Naturalmente, este caso extremo se suele evitar (se intenta resolver por otros medios) ya que encarece sobremanera la aplicación de visión por ordenador.

Las aplicaciones de visión por ordenador intentan, pues, siempre simplificar al máximo el problema, obteniendo sólo los datos que son imprescindibles, y olvidando los demás. Es frecuente que se diseñen para el manejo de objetos rígidos y pesados, que se desean desplazar, almacenar, etc. Sin embargo, existen multitud de sistemas que realizan tareas casi inimaginables, como, por ejemplo, la localización e identificación de los cables de un equipo o componente eléctrico. Un ejemplo mucho más típico, sin embargo, es el sistema CONSIGHT de General Motors, que consiste en líneas generales en una fuente lumi-

nosa que ilumina con un ángulo determinado la cinta transportadora donde se encuentran los objetos. El sensor (la cámara que capta las imágenes) está colocado verticalmente, sobre la cinta, justo encima del punto donde el rayo luminoso incide en ésta. Por este sistema, con el paso de un objeto bajo la luz, el haz se distorsiona de un modo proporcional a la altura del objeto que lo ha distorsionado. Con este sistema, la cámara obtendrá información tanto de la forma del objeto, como de su altura. El CONSIGHT se ha utilizado con bastante éxito para muchas aplicaciones en las que es necesario manejar materiales e identificar su posición y orientación en el espacio.

Otro tipo de aplicaciones de visión por ordenador típicas de la industria se ocupa de controlar movimientos. Las aplicaciones pertenecientes a este segundo grupo intentan resolver dos problemas fundamentales: el control de la trayectoria y evitar colisiones. Al primer grupo de actividades pertenecería, por ejemplo, el caso en el que se «guía» un vehículo o un brazo según una trayectoria especificada mediante marcas que el brazo o el vehículo deben reconocer. La aplicación de control de vehículos que deben circular por una trayectoria determinada es una aplicación muy compleja, y resuelta casi únicamente a nivel de laboratorio, pero día a día se va avanzando. A nivel muy elemental, existen, sin embargo, aplicaciones que realizan este tipo de tareas.

El segundo objetivo de este grupo de aplicaciones de visión artificial consiste «simplemente» en que el sistema evite posibles colisiones del objeto que consideramos con otros objetos, identificados, o sin identificar. Este problema, sin embargo, es más sencillo que el anterior, ya que se trabaja con menos libertad, siendo la trayectoria conocida. Como acabamos de decir, si el entorno de experimentación se encuentra acotado, el problema se simplifica enormemente. De hecho, existen algunos sistemas de este tipo que están funcionando en la industria con bastante éxito.

La complejidad a que puede llegarse con este tipo de aplicaciones es infinita. En estas ocasiones, la velocidad de proceso de las imágenes obtenidas es un factor fundamental. Por ello, de tales imágenes sólo se extraerán aquellas características que sean imprescindibles para obtener la información necesaria. No debemos olvidar que tanto los objetos como la iluminación, como el sensor o la cámara pueden, a su vez, estar moviéndose.

Aplicaciones en control de calidad

Este tipo de aplicaciones tienen un futuro fantástico. Se trata fundamentalmente de comparar el objeto a inspeccionar con un objeto patrón, buscando que concuerde con él con el margen de error establecido. Con ello se obtiene mayor precisión y homogeneidad que si la operación se llevara a cabo por medios humanos, además de eliminar al operario que lleva a cabo este control. La comparación con el objeto patrón se lleva a cabo en la totalidad del objeto, o sólo en aquellas zonas o características que interesen. En el control de calidad, la resolución seleccionada es un factor importante (en ocasiones la imagen debe ser ampliada sustancialmente). Pero al seleccionar la resolución, debe considerarse que si, por ejemplo, ésta excede 1k por línea, será imposible utilizar una memoria de imagen y, por tanto, deberá recurrirse a un scanner o analizador de líneas, cuyo precio, hasta hace muy poco, encarecía enormemente la aplicación. Debemos decir, sin embargo, que los últimos scanners tienen precios mucho más aceptables, por lo que en estas fechas se puede trabajar con excesos de resolución de 2k.

Las aplicaciones para el control de calidad son de enorme importancia. Es frecuente que estén relacionadas con la seguridad. No se trata simplemente de que falle y se rompa

el casco de una botella de vino; pensemos, por ejemplo, en el control de imperfecciones de las cubiertas de caucho de los automóviles, detección de fallas en turbinas de aviones, en la propia estructura del avión, etc. Otras aplicaciones, no relacionadas con la seguridad, pueden ser el control de calidad de tejidos, soldaduras de circuitos, comprobación de filamentos de bombillas, etc. Cada aplicación ha sido diseñada a medida para que pueda acomodarse a las necesidades concretas de cada caso, indicando la precisión admisible, junto con las cualidades o características que se van a analizar.

El ensamblaje de piezas es también otra actividad en la que la visión artificial tiene mucho que hacer. En empresas pequeñas, este ensamblaje y soldado se realiza manualmente, pero en las de gran volumen, las operaciones se realizan automáticamente por máquinas herramientas (robots), que, en muchos casos en los que la acción no es simplemente repetitiva, necesitan de la visión artificial para realizar su tarea. Los sistemas existentes son muy caros y complejos. De hecho, es una de las tareas que requiere procesar más información (control del movimiento, inspección, manejo etcétera). Estas aplicaciones, por su enorme complejidad, siguen perfeccionándose en los laboratorios de desarrollo. Sin embargo, los grandes fabricantes de automóviles disponen ya de sistemas de este tipo para pequeños ensamblajes (motores y bombas eléctricas, etc.). Dada la complejidad de los sistemas (requieren varias cámaras, algunas de las cuales suelen ser móviles, otros tipos distintos de sensores-táctiles, etc.) sólo grandes compañías pueden permitirse disponer de ellos, siendo además, por el momento, menos útiles que en otras aplicaciones.

Terminaremos indicando que la tendencia del mercado de aplicaciones de visión artificial o visión por ordenador es simplificar al máximo los requerimientos. Es necesario estudiar exhaustivamente la tarea a realizar, para no enterrar recursos en algo a lo que a penas si se le va a sacar partido. Las aplicaciones de este tipo requieren el proceso de mucha información, y todos los recortes que se puedan realizar en ésta serán en provecho de la propia aplicación.

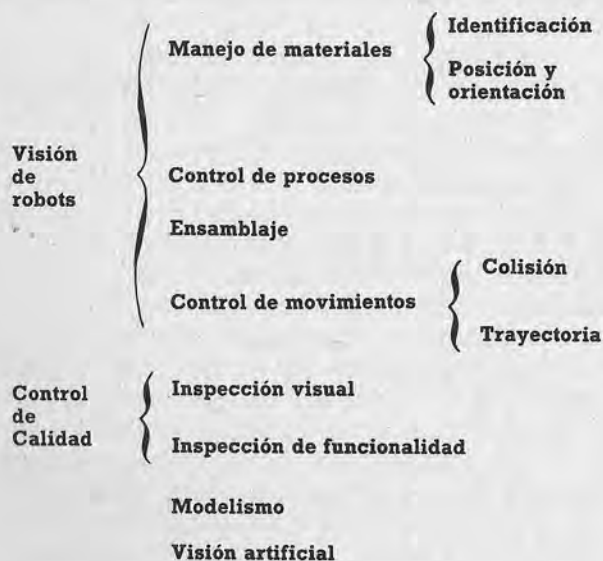


Fig. 2

TERMINOLOGIA

Algoritmo. Conjunto de reglas cuya aplicación permite llegar a un objetivo computacional dado. Por ejemplo, puede utilizarse un algoritmo para eliminar las líneas o las superficies ocultas, relleno de superficies, eliminando de ruidos de fondo, realzado de bordes, etc.,

Alisamiento. En gráficos, cuando el dispositivo de presentación intenta llegar a detalles que exceden el poder de resolución del sistema. Suele llamarse también efecto escalera, ya que los bordes no aparecen rectos, sino formando líneas con escalones. Este efecto es mucho más evidente en sistemas de baja resolución: en los sistemas con mayor definición puede eliminarse mediante técnicas anti-alisamiento.

Animación. Técnicas para conseguir un cambio o movimiento de forma continua, imitando el modo en que se produce en el mundo real. Estas técnicas utilizan conjuntos de imágenes muy semejantes que se proyectan a una velocidad tal que el ojo humano no puede percibir el cambio de una a otra, y el movimiento le parece continuo. Los proyectores de cine de 8 mm. suelen ser de 18 ó 24 imágenes por segundo, y los de 16 mm. siempre proyectan a 24 imágenes por segundo, velocidad que ya ha sido adoptada por la práctica totalidad de las casas fabricantes.

Antialisamiento. Técnica de localización de los errores de alisamiento introducidos por los sistemas discretos. Así, los bordes escalonados se pueden suavizar filtrando las in-

tensidades de los pixels donde se produce la discontinuidad, para que los cambios de color resulten más suaves.

Brillantes, superficies. Superficies que resultan altamente reflectantes de la luz. Este efecto se puede conseguir mediante la inclusión de un parámetro "de brillo", algoritmos de cálculo del haz de luz que se refleja sobre una superficie.

Brillo. Está relacionado con la intensidad luminosa. Esta es directamente proporcional a la potencia de la fuente de luz e inversamente al cuadrado de la distancia del observador a dicha fuente luminosa. Sin embargo, el brillo, aunque está relacionado con la intensidad luminosa, depende a su vez de la zona iluminada, y su orientación. Es frecuente suponer (para abordar un problema de visión por computador) que las superficies convencionales reflejan la luz del mismo modo en cualquier dirección del espacio, y también que la fuente luminosa está suficientemente alejada del objeto problema como para que los cambios en sus distintas zonas son prácticamente insignificantes. Para averiguar el brillo existente en un punto cualquiera con las suposiciones anteriores, basta saber la intensidad del foco luminoso, y el ángulo de incidencia del haz de luz.

Cara oculta. Siempre que exista una colección de objetos, visualizaremos un conjunto de caras de dichos objetos. Sin embargo, las leyes físicas nos impiden ver todas las facetas. En muchos casos, estos lados ocultos no

TERMINOLOGIA

interesan en la imagen, por lo que se puede evitar dibujarlos. Para eliminar dichas caras, se aplica un algoritmo determinado, obtenido a partir de la ecuación del plano de dicha superficie, o mediante el vector normal a dicho plano.

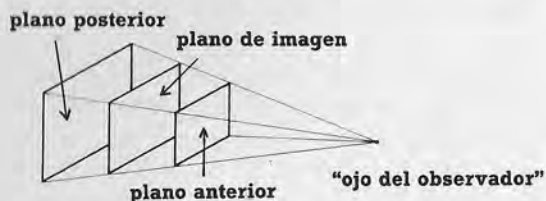


Fig. 1. Volumen de recorte ("clipping volume") para una representación tridimensional (es el tronco de la pirámide comprendido entre los planos anterior y posterior).

Clipping (recorte). Eliminación de los elementos de la imagen que "caen" fuera del "volumen de recorte" ("clipping volume") formado en la pirámide de visión por los planos anterior y posterior del campo a representar. (Véase dibujo).

Coherencia. Cuando se aplica un algoritmo para eliminar las caras ocultas, éste puede aplicarse por ejemplo dividiendo la imagen en tiras horizontales, y analizando las condiciones de dicha tira (intersecciones, etc.). Una vez se ha detectado que un determinado objeto tapa a otro, no es necesario realizar los cálculos para otras tiras siguientes.

Cono de visión. Campo visual. Se considera un punto fijo donde supuestamente se encuentra el ojo del espectador. Por la estructura del ojo humano, sólo se pueden ver objetos que se encuentren en un ángulo de 60 grados por encima y debajo de la línea visual (la que va del ojo al punto central observado). Por tanto, sólo se puede ver (sin distorsión) los objetos que se abarcan con un ángulo de 100 a 120 grados con vértice en el "ojo del observador" este es el cono de visión. Naturalmente, en la pantalla plana del dispositivo de presentación visual no existe esta limitación, pero hay que suponerla y tenerla en cuenta si se quieren presentar imágenes realistas.

Convexos, polígonos. Polígonos (formas cerradas por líneas rectas) cuyos ángulos internos entre dos líneas consecutivas son menores que 180 grados.

Coordenadas desde el observador. Las escenas tridimensionales pueden plasmarse fácilmente creando un sistema de coordenadas asociado a la posición del observador. Para trazar los ejes, basta conocer la posición del observador y tomar otro punto de observación cualquiera.

Cromaticidad. Si se examinan las distintas longitudes de onda correspondientes a un determinado color, podrá observarse que algunas son más intensas. La cromaticidad es el parámetro que describe la longitud de onda dominante para un determinado color, en otras palabras, su pureza.

Difusa, iluminación. Existen técnicas especiales para simular una iluminación difusa. Es un sistema para representar una iluminación que proviene de más de una fuente luminica. La cantidad de luz difusa que puede ver un observador depende de la intensidad de la fuente, el ángulo que forma con la superficie y del coeficiente de reflectancia de esta superficie. La posición u orientación relativa del observador no influyen en absoluto, por lo que este tipo de iluminación produce un efecto "plano" que generalmente es necesario mejorar mediante reflexiones especulares.

Digitalizador. Dispositivo que convierte las imágenes analógicas en una información organizada de forma y con valores digitales. El digitalizador divide la imagen en puntos, y envía una señal por cada uno de ellos (representativa de la intensidad de luz existente en ese punto para cada uno de los colores básicos, si la imagen es en color).

Digitalizar. Convenir una imagen en una matriz de valores a los que se encuentra asociada determinada información (la intensidad o intensidades en cada punto).

Eliminación de las líneas ocultas. La información relativa a una imagen se almacena en forma de una matriz de valores, o mediante algún algoritmo o ecuación. Sin embargo, al representarla, aparecerán tanto los puntos que se ven, como aquellos que deberían aparecer "tapados". Para ello se utilizan algoritmos, que miden el ángulo formado por la línea de visión del observador y el vector normal a la cara del objeto. Si este ángulo es superior a 90 grados, el ob-

jeto no debe ser visible. Con este criterio, se deberán analizar todos los vértices y aristas, y si el objeto es irregular, ha de tenerse en cuenta a la hora de aplicar el algoritmo). (Véase dibujo).

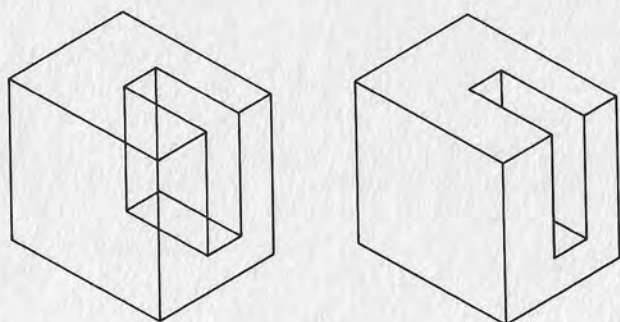


Fig. 2. A la derecha se muestra el resultado de haber eliminado las "líneas ocultas" de la imagen de la izquierda.

Eliminación de superficies ocultas. Técnicas que eliminan de la imagen aquellas superficies que el observador no puede ver. Véase eliminación de líneas ocultas. Para la eliminación de superficies ocultas, existen numerosos algoritmos, algunos más sencillos y menos eficaces, otros más complejos pero de mejores resultados. (Véase dibujo).

Grabador de imágenes. Técnica de preparación de las imágenes de vídeo para su proceso mediante ordenador. Para ello se conecta una cámara de vídeo a un digitalizador que convierte la señal analógica de la cámara en señales digitales... En un momento dado, la señal digital puede mantenerse o "grabarse" y, ser manipulada a continuación.

Holograma. Técnica por la que se ilumina la escena mediante láser, y se hace que la luz reflejada impresione una placa fotográfica iluminada a su vez, con la luz original. Los patrones de las interferencias que se crean, forman el holograma.

Memoria de imágenes (raster). Dispositivo que es capaz de almacenar una imagen monocroma o en color en forma de información de los pixels que la componen (imagen digital). En otro momento, mediante otro dispositivo, la señal digital puede ser convertida en señal analógica y visualizada en la pantalla.

Parpadeo. Este se produce cuando la pantalla no se renueva a una velocidad compatible con la persistencia de las imágenes en la retina del ojo humano. Esta persistencia de las imágenes es la que permite ver en el cine y televisión las imágenes continuas, no una tras otra... No existe ninguna velocidad óptima para el cambio de las imágenes, pues experimentos en este campo han mostrado que depende del brillo, color, y contrastes existentes en la imagen, entre otros factores.

Pirámide de visión. Es la formada tomando como vértice el punto de vista del observador y como base rectangular el espacio (plano) donde se proyectan las imágenes.

Razón de aspecto. Relación entre la altura y anchura de una imagen. Para televisión, esta relación suele ser de 3:4. En cine, puede variar, dependiendo del formato, pero para películas de 35 mm. de 2:3.

